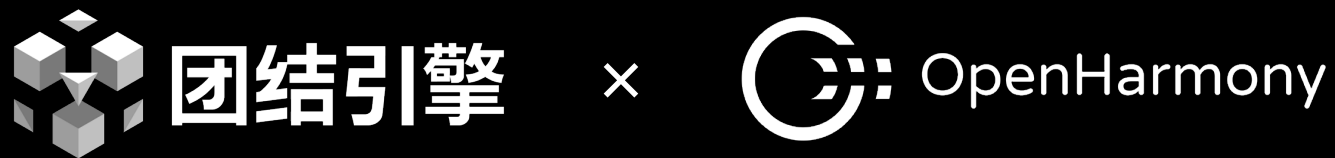




全面支持OpenHarmony 团结引擎共建游戏新生态

2023

刘伟贤 Unity中国 高级技术经理





HarmonyOS

OpenHarmony

鸿蒙底层内核系统
集成Linux内核+LiteOS
具备底层通信能力

已开源

HarmonyOS

基于OpenHarmony
和安卓 (AOSP) 打造的
手机系统 包含UI界面

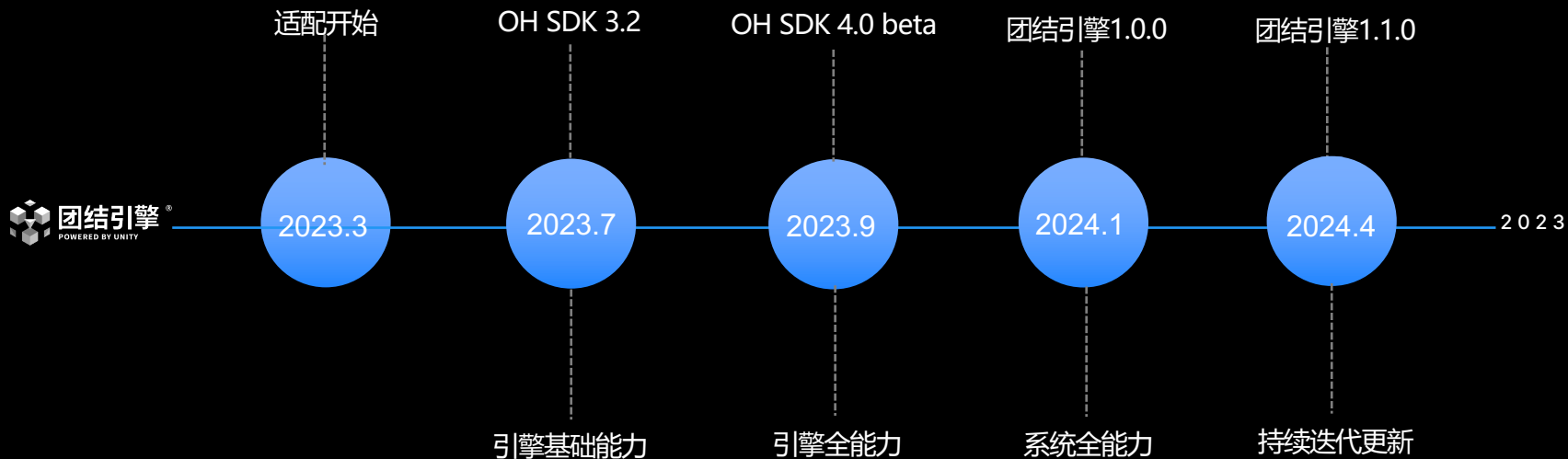
未开源

Harmony OS NEXT

在HarmonyOS基础上
剔除安卓 (AOSP) 后的
产品, 属于全新系统

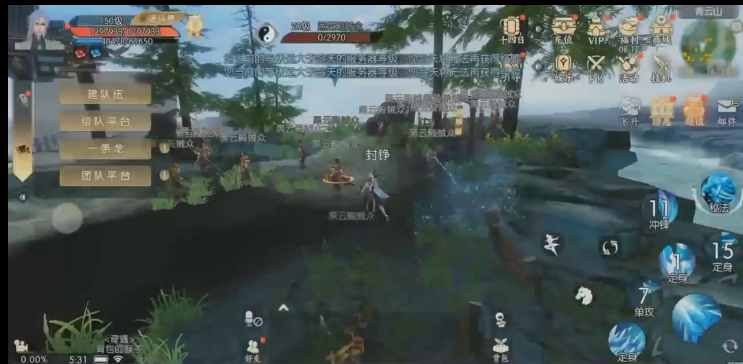
未开源

OpenHarmony 适配时间轴





部分客户适配成果



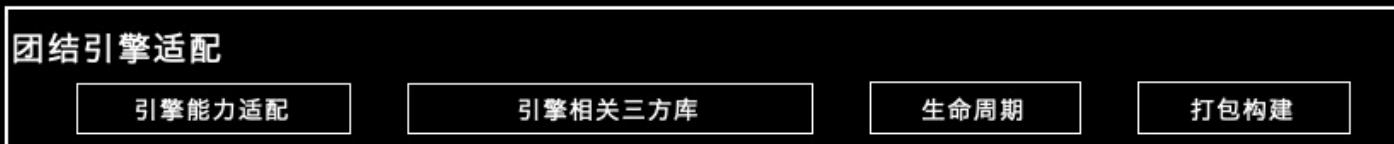


OpenHarmony适配流程



研发商工作:

1. 游戏适配, 包括游戏主体逻辑适配、三方库适配、兼容性适配;
2. 中台适配, 包括登录、支付、广告等;



Unity中国工作:

1. 引擎能力适配, 包括Editor、工具链等;
2. 闭源库适配;
3. 生命周期适配;
4. 打包构建适配;



系统工作:

1. Stage模型适配、XComponent适配、多媒体适配、NAPI适配;
2. 性能优化;



OpenHarmony-环境配置

2023



OpenHarmony SDK下载

<https://gitee.com/openharmony/docs/blob/master/zh-cn/release-notes/OpenHarmony-v4.0-release.md#%E9%85%8D%E5%A5%97%E5%85%B3%E7%B3%BB>

从镜像站点获取

表2 获取源码路径

版本源码	版本信息	下载站点	SHA256校验码	软件包容量
全量代码（标准、轻量和小型系统）	4.0 Release	站点	SHA256校验码	27.6 GB
Hi3861解决方案（二进制）	4.0 Release	站点	SHA256校验码	27.3 MB
Hi3516解决方案-LiteOS（二进制）	4.0 Release	站点	SHA256校验码	302 MB
Hi3516解决方案-Linux（二进制）	4.0 Release	站点	SHA256校验码	194 MB
RK3568标准系统解决方案（二进制）	4.0 Release	站点	SHA256校验码	5.1 GB
标准系统Public SDK包（Mac）	4.0.10.13	站点	SHA256校验码	841 MB
标准系统Public SDK包（Mac-M1）	4.0.10.13	站点	SHA256校验码	797 MB
标准系统Public SDK包（Windows/Linux）	4.0.10.13	站点	SHA256校验码	2.0 GB



DevEcoStudio SDK下载

Preferences

SDK

OpenHarmony

Location: /Users/liuweixian/Library/OpenHarmony/Sdk [Edit](#)

The following are SDKs required for OpenHarmony app development. DevEco Studio will notify you when updates are available.

Name	Version	Stage	Status
✓ API Version 10			
✓ ArkTS	4.0.10.16	Release	Installed
✓ JS	4.0.10.16	Release	Installed
✓ Native	4.0.10.16	Release	Installed
✓ Previewer	4.0.10.16	Release	Installed
✓ Toolchains	4.0.10.16	Release	Installed

Hide Obsolete Packages

Cancel Apply OK



Editor配置

OpenHarmony

OpenHarmony SDK Installed with Tuanjie (recommended)

OpenHarmony SDK

/Users/liuweixian/Library/OpenHarmony/Sdk/10

Browse



You are not using the recommended OpenHarmony SDK, other versions are not officially supported.

Node.js SDK Installed with Tuanjie (recommended)

Node.js

/Users/liuweixian/nodejs

Browse



You are not using the recommended Node.js SDK, other versions are not officially supported.

JDK Installed with Tuanjie (recommended)

JDK

/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/

Browse

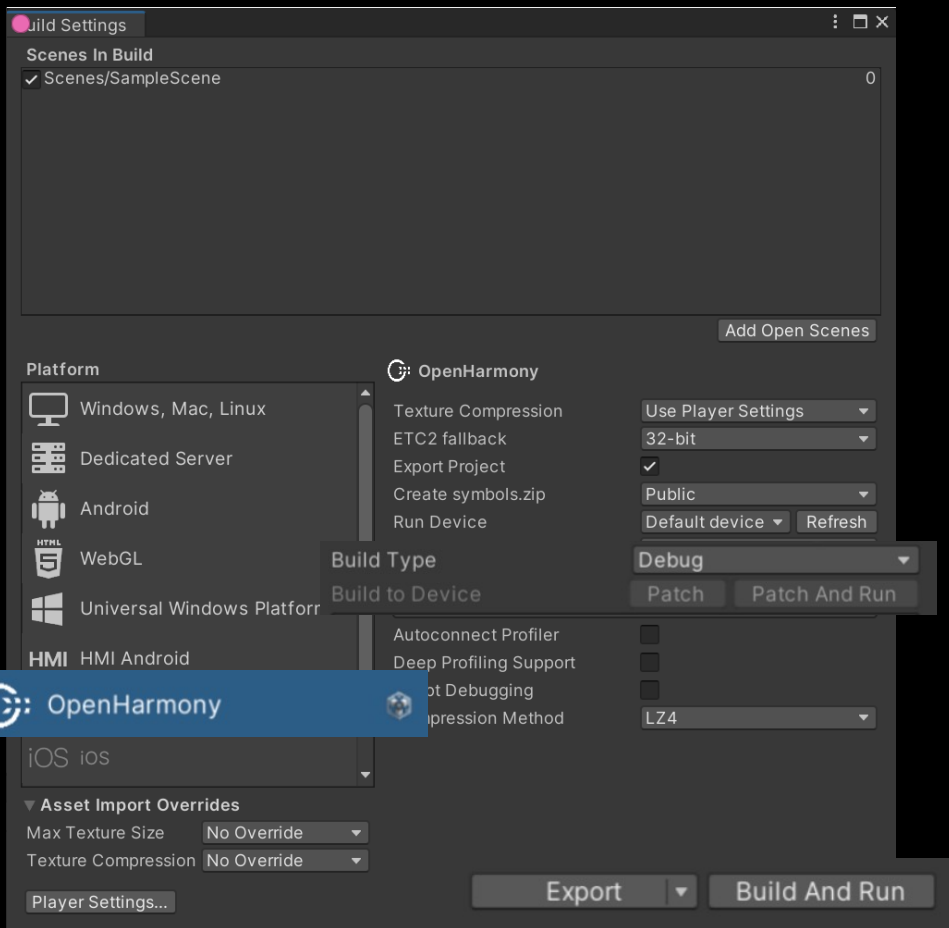


You are not using the recommended JDK, other versions are not officially supported.



Editor workflow-OpenHarmony

2023



与原有工作流完全一致
兼容老版本项目

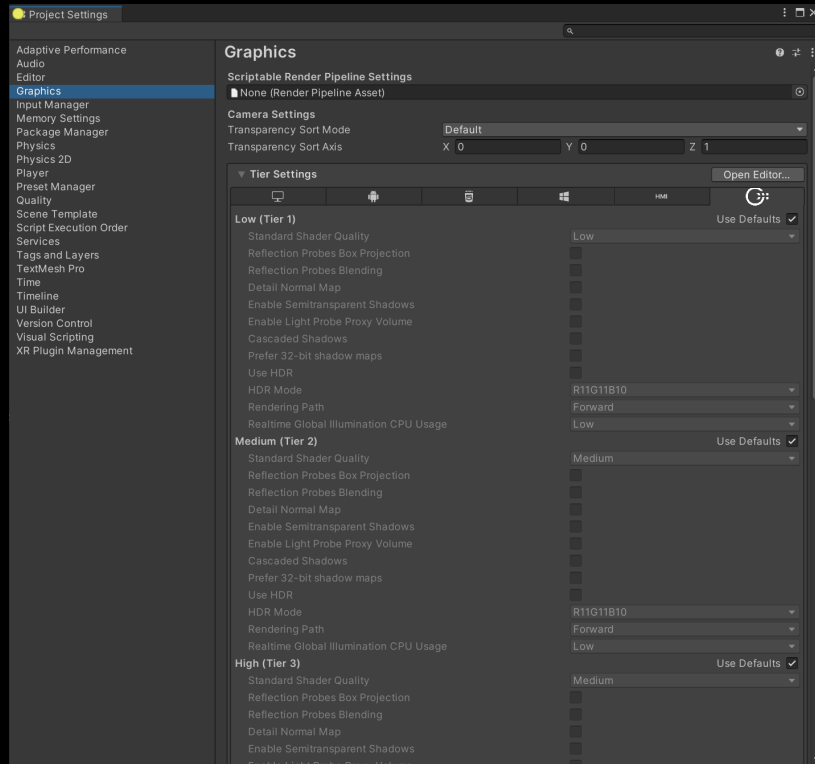
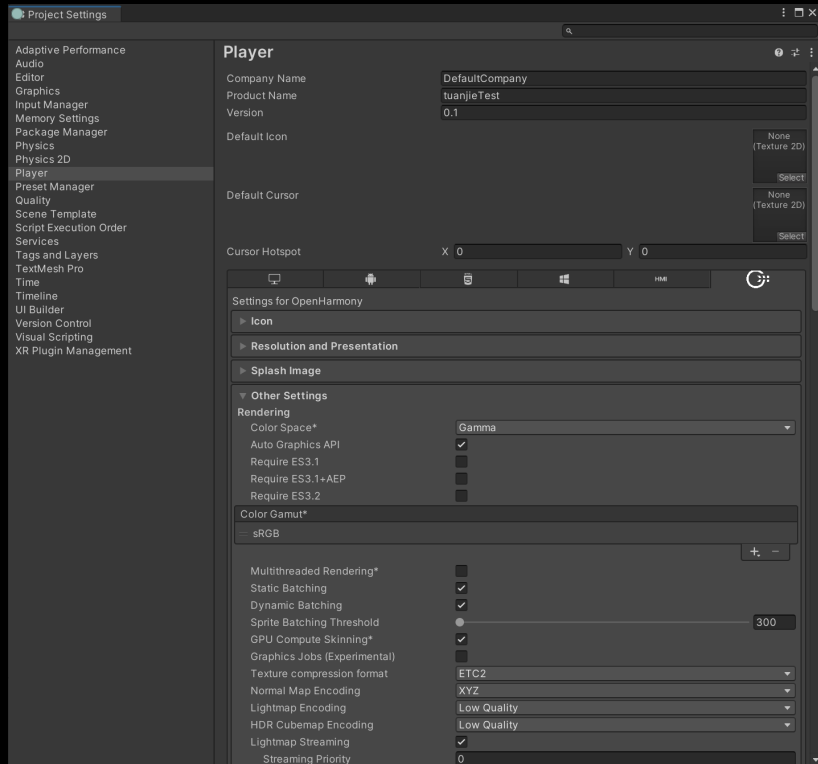
支持导出Deveco Studio原生Ability Project
支持一键出包Hap

支持连接设备Patch And Run
支持连接设备Build And Run





Openharmony平台专有的配置项





Deveco Studio Ability Project

The screenshot displays the DeVeCo Studio IDE interface. The main window shows the project structure for 'L10DevEco'. The 'resources' directory is expanded, showing sub-directories like 'base', 'en_US', and 'rawfile'. The 'Data' folder is also expanded, listing various asset files such as 'boot.config', 'globalgamemangers', and 'resources.assets'. The 'TuanjiePlayerAbility.ts' file is highlighted in the 'ability' directory.

On the right side, the 'Notifications' panel shows two messages: 'Sync project finished in 2 s 624 ms' and 'Sync project started', both timestamped at 10:21. Below the notifications, there is a search bar with the text 'Search Everywhere Double ⌵', a 'Go to File' button with a magnifying glass icon, 'Recent Files' with a keyboard shortcut '⌘E', and a 'Navigation Bar' with a keyboard shortcut '⌘1'. A message 'Drop files here to open them' is displayed below these options.

The bottom status bar shows the log level 'Log: HILog FaultLog' and the project ID 'PNM0123316000056'. The bottom right corner of the IDE has a 'Verbose' dropdown and an 'updateDisplayInfo' button.



OpenHarmony Device Connector

- hdc shell
- hdc install
- hdc uninstall
- hdc list target
- hdc kill
- hdc start
- hdc file send
- hdc file recv
- hdc hilog



引擎能力-OpenHarmony

2023



整体架构

- 引擎线程与UI线程分离
 - 在ts创建worker.ThreadWorker
 - 引擎Loop通过libuv事件驱动
- 支持单/多线程渲染
- 支持Job System/C# Job System



渲染能力

- 支持内置渲染管线及URP
- 支持OpenGL ES 3.x, Vulkan(适配完成, 测试中)
- 纹理压缩格支持与安卓一致, 默认使用ETC2



文件系统

- 文件也为包内文件与包外文件，与安卓一样
 - Openharmony安装包为hap，也是一个压缩包，包内文件都被压缩hap安装后也是压缩包形式，类似apk安装后有package/base.apk
 - 文件读写与安卓一样，C# File文件无法读写包内文件，WWW,UnityWebRequest,Assetbundle除外
- Application底下的路径变量都映射OpenHarmony底下的应用沙箱路径



音/视频能力

- 音频
 - 引擎使用OpenSL ES
 - 第三方Wwise插件或者原生Fmod插件需自行重新编译
- 视频支持原生ts接口及硬解码
 - ts层提供VideoPlayer组件直接使用原生ts接口播放视频
 - native层VideoPlayer通过硬解码获得视频画面由于引擎渲染
 - 第三方插件需自行重新编译



脚本能力

- C#与ts交互
 - C# Call到Native层, Napi支持
 - .tslib文件用于设置导出对象给C#使用
 - export function Register<FileName>()
 - .ts/.ets文件用于作为Plugins的代码直接带入到原有项目

TestClass.tslib

```
export function RegisterTestClass() {  
  var register = {}  
  register["StaticClassTest"] = StaticClassTest;  
  register["ClassObjectTest"] = new ClassObjectTest();  
  register["deviceInfo"] = deviceInfo;  
  register["sensor"] = sensor;  
  return register;  
}
```



脚本能力

- C#与ts交互
 - C#侧有OpenHarmonyJSClass，用于直接调用Static Function
 - C#侧有OpenHarmonyJSObject，用于ts导出对象的表示

TestClass.tslib

```
export function RegisterTestClass() {  
    var register = {}  
    register["StaticClassTest"] = StaticClassTest;  
    register["ClassObjectTest"] = new ClassObjectTest();  
    register["deviceInfo"] = deviceInfo;  
    register["sensor"] = sensor;  
    return register;  
}
```

```
OpenHarmonyJSClass openHarmonyJSClass = new OpenHarmonyJSClass("StaticClassTest");  
openHarmonyJSClass.CallStatic("voidFunc");
```

```
sensorJSObject = new OpenHarmonyJSObject("sensor");  
sensorOnCallback = new OpenHarmonyJSCallback(SensorOnCallback);  
sensorJSObject.Call("on", 1, sensorOnCallback);
```



脚本能力

- C#与ts交互
 - C#侧支持导出后直接调用ts的系统api
 - C#侧提供OpenHarmonyJSCallback，用于支持异步接口回调

TestClass.tslib

```
export function RegisterTestClass() {  
    var register = {}  
    register["StaticClassTest"] = StaticClassTest;  
    register["ClassObjectTest"] = new ClassObjectTest();  
    register["deviceInfo"] = deviceInfo;  
    register["sensor"] = sensor;  
    return register;  
}
```

```
sensorJSObject = new OpenHarmonyJSObject("sensor");  
sensorOnCallback = new OpenHarmonyJSCallback(SensorOnCallback);  
sensorJSObject.Call("on", 1, sensorOnCallback);
```

```
public void SensorOnCallback(params OpenHarmonyJSObject[] args)  
{  
    OpenHarmonyJSObject data = args[0];  
    Debug.Log("Sensor:" + data.Get<float>("x") + "," + data.Get<float>("y") + "," + data.Get<float>("z"));  
}
```



脚本能力

- C#与ts交互
 - C#侧提供OpenHarmonyJSArrayBuffer用于支持ts的ArrayBuffer

```
OpenHarmonyJSClass JsClass = new OpenHarmonyJSClass("ArrayBufferTest");
OpenHarmonyJSObject fs = new OpenHarmonyJSObject("fs");

int byteNum = 2;
string filepath = Application.persistentDataPath + "/Test.bin";
OpenHarmonyJSArrayBuffer arrayBufferWrite = new OpenHarmonyJSArrayBuffer(byteNum);
OpenHarmonyJSArrayBuffer arrayBufferWrite2 = new OpenHarmonyJSArrayBuffer(byteNum);
OpenHarmonyJSArrayBuffer arrayBufferRead = new OpenHarmonyJSArrayBuffer(byteNum);

int fileId = JsClass.CallStatic<int>("OpenFile", filepath);

byte[] nums = new byte[] { 60, 70 };
byte[] nums2 = new byte[] { 50, 60 };
arrayBufferWrite.SetValue(nums);
arrayBufferWrite2.SetValue(nums2);

//Write single arraybuffer
fs.Call("writeSync", fileId, arrayBufferWrite);
fs.Call("closeSync", fileId);

//Read single arraybuffer
fileId = JsClass.CallStatic<int>("OpenFile", filepath);
fs.Call("readSync", fileId, arrayBufferRead);
fs.Call("closeSync", fileId);
```




Profiler能力

- 内置的Profiler能力与原来一样，真机可用
- 平台对应的Profiler及抓帧工具—SmartPerf

SmartPerf Editor

SmartPerf游戏性能调优工具

全平台游戏性能测试，助力解决游戏测试过程中的各种问题，协助游戏企业高效推出高质量受欢迎的游戏产品。

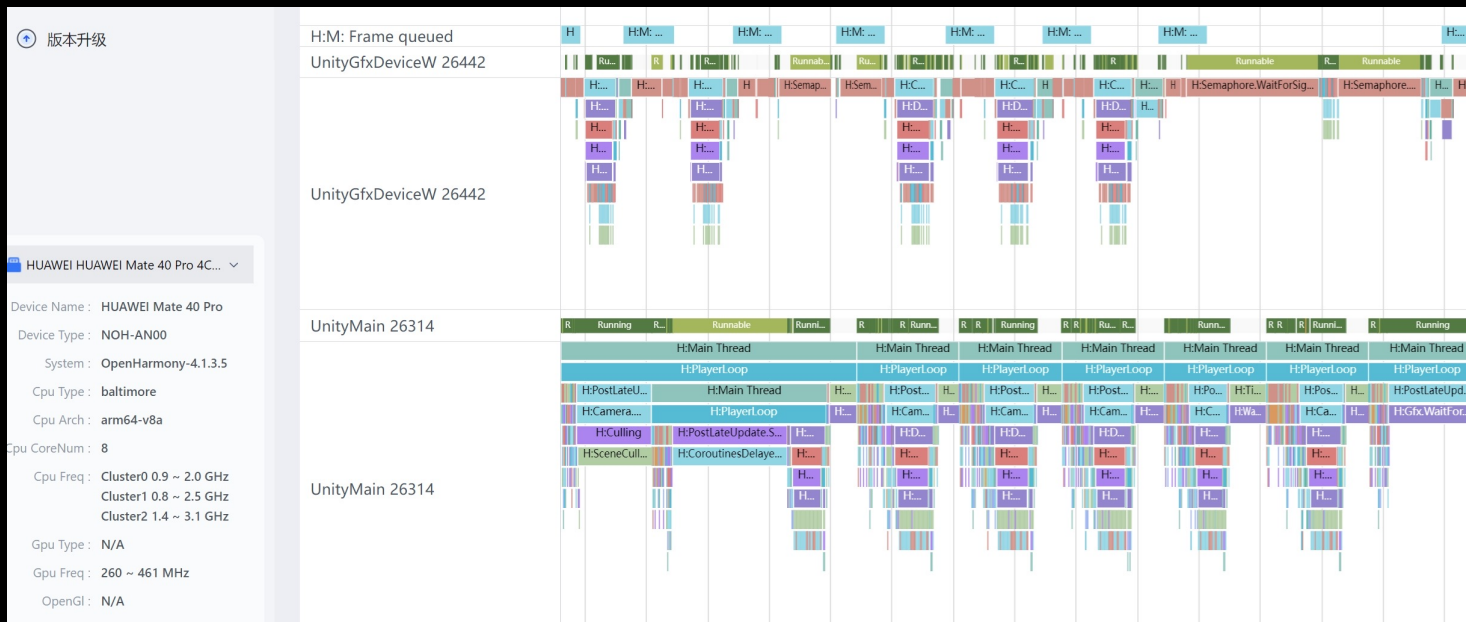
[开始体验 >](#)

- 支持多平台多终端**
支持HarmonyOS/Android平台，支持多类型终端，如手机、平板等。
- 无需ROOT**
设备无需ROOT，高效解决HarmonyOS和Android性能测试分析难题。
- 测试数据准确全面**
工具本身对测试设备CPU的性能基本无影响，轻松获得FPS、Jank、CPU、GPU、Memory、Power、Temp、Network等性能参数。
- 工具易用性高**
工具安装快捷，操作简单，减少繁琐的测试障碍，不仅支持游戏，普通应用也可使用。



Profiler能力

- SmartPref中hitrace支持, unity profiler tag直接输出, 在development build下默认生效





系统相关

- 权限相关获取

```
string[] permissions = new string[]
{
    Permission.CoarseLocation,
    Permission.FineLocation,
    Permission.Camera,
    Permission.Microphone,
    Permission.ExternalStorageRead,
    Permission.ExternalStorageWrite
};
PermissionCallbacks callbacks = new PermissionCallbacks();
callbacks.PermissionGranted += (string permission) =>
{
};
callbacks.PermissionDenied += (string permission) =>
{
};
Permission.RequestUserPermissions(permissions, callbacks);
```



系统相关

- 位置服务
- Sensor, 陀螺仪
- 屏幕息屏
 - `Screen.sleepTimeout = SleepTimeout.NeverSleep;`
- Webview支持
- `OpenUrl`
- 等等



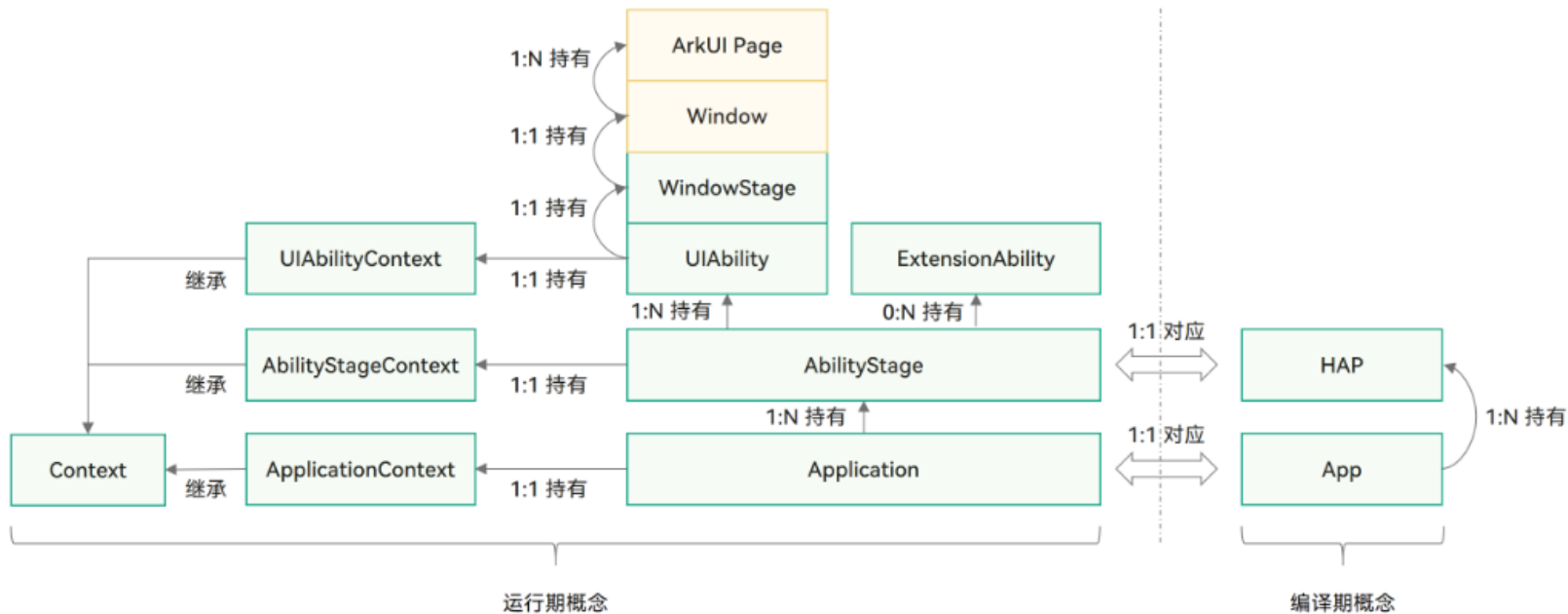
团结引擎适配技术细节

2023



OpenHarmony Staging模型

图1 Stage模型概念图



来源: <https://docs.openharmony.cn/pages/v4.0/zh-cn/application-dev/application-models/stage-model-development-overview.md/>



OpenHarmony ArkUI & XComponent

```
build() {  
  Stack() {  
    XComponent({ id: 'TuanjiePlayer', type: 'surface', libraryname: 'tuanjie' })  
      .onLoad((context) => {  
        TuanjieLog.info('%{public}s', 'XComponent onLoad');  
      })  
      .onDestroy(() => {  
        TuanjieLog.info('%{public}s', 'XComponent onDestroy');  
      })  
      .onAppear(() => {  
        TuanjieLog.info('%{public}s', 'XComponent onAppear');  
      })  
      .onDisAppear(() => {  
        TuanjieLog.info('%{public}s', 'XComponent onDisAppear');  
      })  
      .width('100%')  
      .height('100%')  
  
    TuanjieWebView({viewInfo: this.webviewInfo});  
    StaticSplashScreen();  
    VideoPlayer();  
  }  
}
```

XComponent¹⁰⁺

XComponent(value: {id: string, type: XComponentType, libraryname?: string, controller?: XComponentController})

参数:

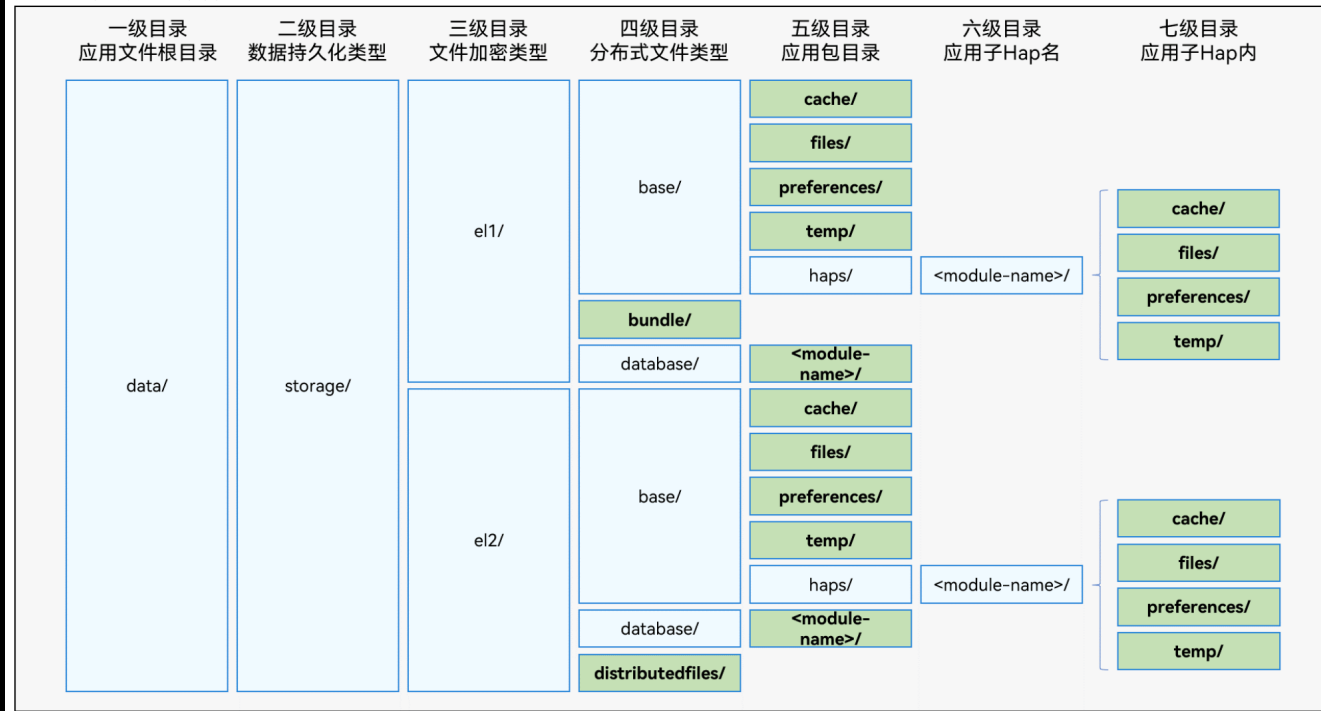
参数名	参数类型	必填	描述
id	string	是	组件的唯一标识, 支持最大的字符串长度128。
type	XComponentType	是	用于指定XComponent组件类型。
libraryname	string	否	用Native层编译输出动态库名称, 仅类型为SURFACE或TEXTURE时有效。
controller	XComponentcontroller	否	给组件绑定一个控制器, 通过控制器调用组件方法, 仅类型为SURFACE或TEXTURE时有效。

来源: <https://docs.openharmony.cn/pages/v4.0/zh-cn/application-dev/reference/arkui-ts/basic-components-xcomponent.md>



RawFile & 应用沙箱

图3 应用文件目录结构图



来源: <https://docs.openharmony.cn/pages/v4.0/en/application-dev/file-management/app-sandbox-directory.md/>



Napi

- 在OpenHarmony中，C API中的N-API接口可以实现ArkTS/TS/JS与C/C++之间的交互。N-API提供的接口名与三方Node.js一致，目前支持部分接口
- ArkTS/TS/JS侧通过import引入native侧的so文件，如：import hello from 'libhello.so'，意为使用libhello.so的能力，并将名为hello的ArkTS/TS/JS对象给到应用的ArkTS/TS/JS侧，开发者可通过该对象，调用到在cpp中开发的native方法



OpenHarmony Roadmap

- SDK升级 API11支持
 - 文件系统改动, 目前超过2G的文件会出现读卡死的情况
 - 系统回调更新, 切后台无法暂停应用
- OH_Audio支持
 - 游戏音频和音乐不能同时播放
- 其他Bug修复



Thank you

2023