**诚鸿OS\_IDE环境配置**

**诚鸿OS(ChengHongOS)**

目录

[1 概述 3](#_Toc20355)

[2 开发环境配置要求 3](#_Toc29672)

[2.1 开发环境 3](#_Toc30814)

[2.2 调试与演示环境 3](#_Toc3769)

[3 开发环境 4](#_Toc13922)

[3.1 安装目录配置 4](#_Toc15722)

[3.2 安装DevEco Studio 4](#_Toc13243)

[3.3 配置中文环境 5](#_Toc24126)

[3.4 安装SDK 7](#_Toc26432)

[3.5 配置ArtTS开发环境 10](#_Toc24526)

[3.1.1 创建ArtTS项目 10](#_Toc29657)

[3.1.2 修改项目配置 12](#_Toc21830)

[3.1.3 导入产品兼容性ID 12](#_Toc21573)

[3.1.4 编写页面 14](#_Toc14759)

[4 源码工程（应用级）加签 14](#_Toc15826)

[5 应用上传 15](#_Toc16914)

[5.1 hdc连接设备 15](#_Toc14846)

[5.2 IDE上传和运行 16](#_Toc24419)

[6 配置Qt开发环境与开发 17](#_Toc13866)

[6.1 配置Qt开发环境 17](#_Toc20899)

[6.2 编写Qt项目 26](#_Toc443)

# 概述

本文档主要介绍诚迈鸿志HongZOS桌面操作系统的软件开发环境配置方法，包含：硬件需求、软件需求、IDE安装配置、SDK配置、原生程序的工程配置和调试方法、QT程序的工程配置和调试方法。

在诚迈鸿志HongZOS桌面操作系统环境下，用户可以根据此文档进行开发环境配置、程序开发、程序调试、打包构建等操作。本文档适用于鸿蒙领域的开发者。

# 开发环境配置要求

## 开发环境

| **序号** | **名称** | **配置要求** |
| --- | --- | --- |
| 1 | 操作系统 | Windows10 64位或macOS 10.14/10.15/11.2.2 |
| 2 | 内存 | 8GB及以上 |
| 3 | 硬盘 | 100GB及以上 |
| 4 | 分辨率 | 1280\*800像素及以上 |

## 调试与演示环境

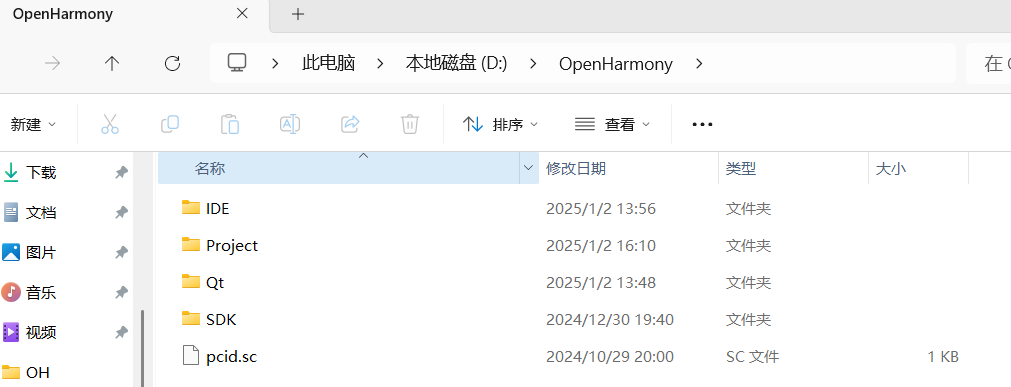
以诚迈鸿蒙电脑作为调试设备，配置如下：

| **序号** | **名称** | **配置要求** |
| --- | --- | --- |
| 1 | 操作系统 | 诚鸿OS(ChengHong OS) |
| 2 | CPU | 3A6000 2.5GHz |
| 3 | 内存 | 8GB及以上 |
| 4 | 硬盘 | 100GB及以上NVME硬盘 |
| 5 | 分辨率 | 1920\*1080像素及以上 |

# 开发环境

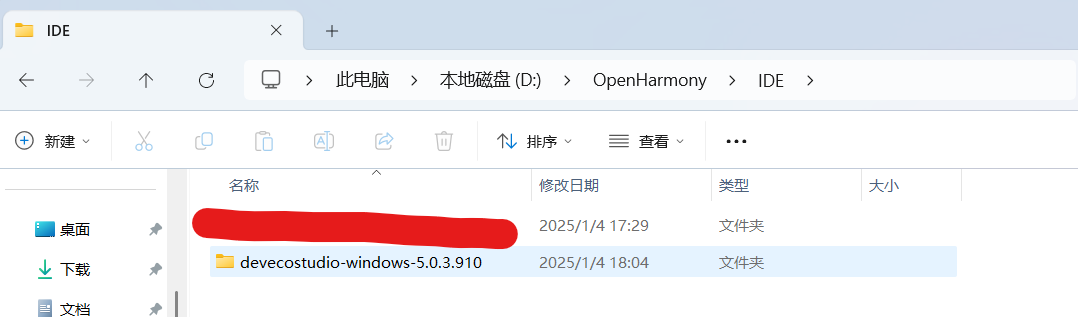
## 安装目录配置

选择C盘或者D盘新建目录OpenHarmony，例如：D:\OpenHarmony。在OpenHarmony下新建子目录IDE，SDK，Qt，Project。

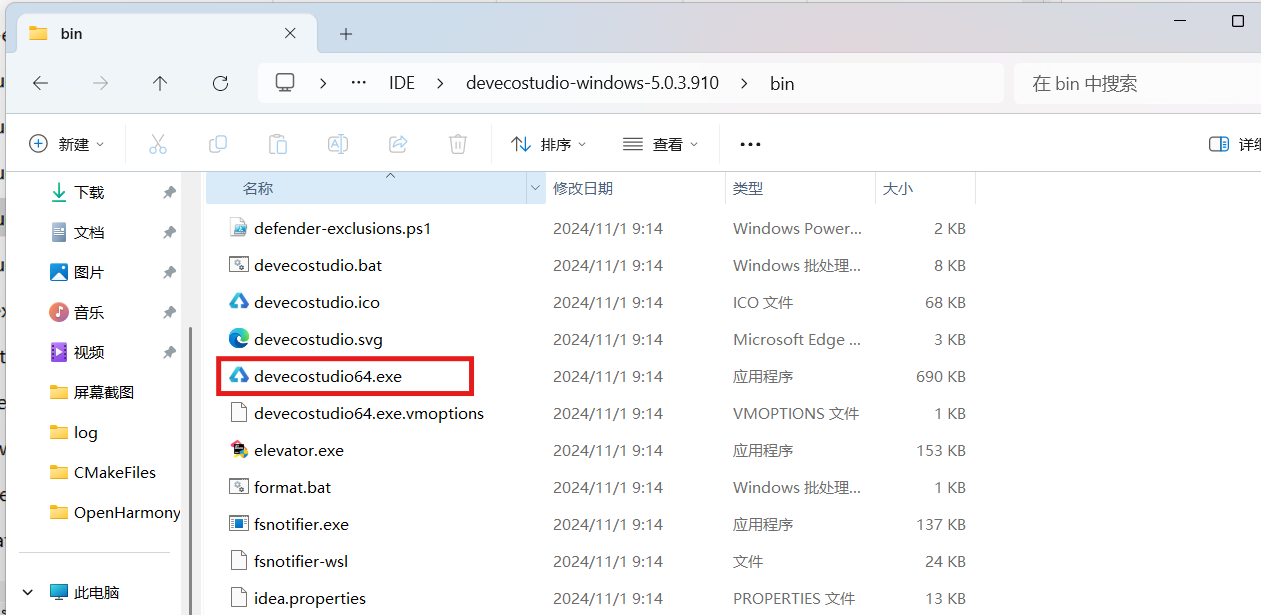


## 安装DevEco Studio

将 OpenHarmony-5.0.0-release\devecostudio-windows-5.0.3.910.zip解压到D:\OpenHarmony\IDE目录下。



D:\OpenHarmony\IDE\devecostudio-windows-5.0.3.910\bin目录下devecostudio64.exe为程序入口，可发送到桌面作为程序的打开快捷键。

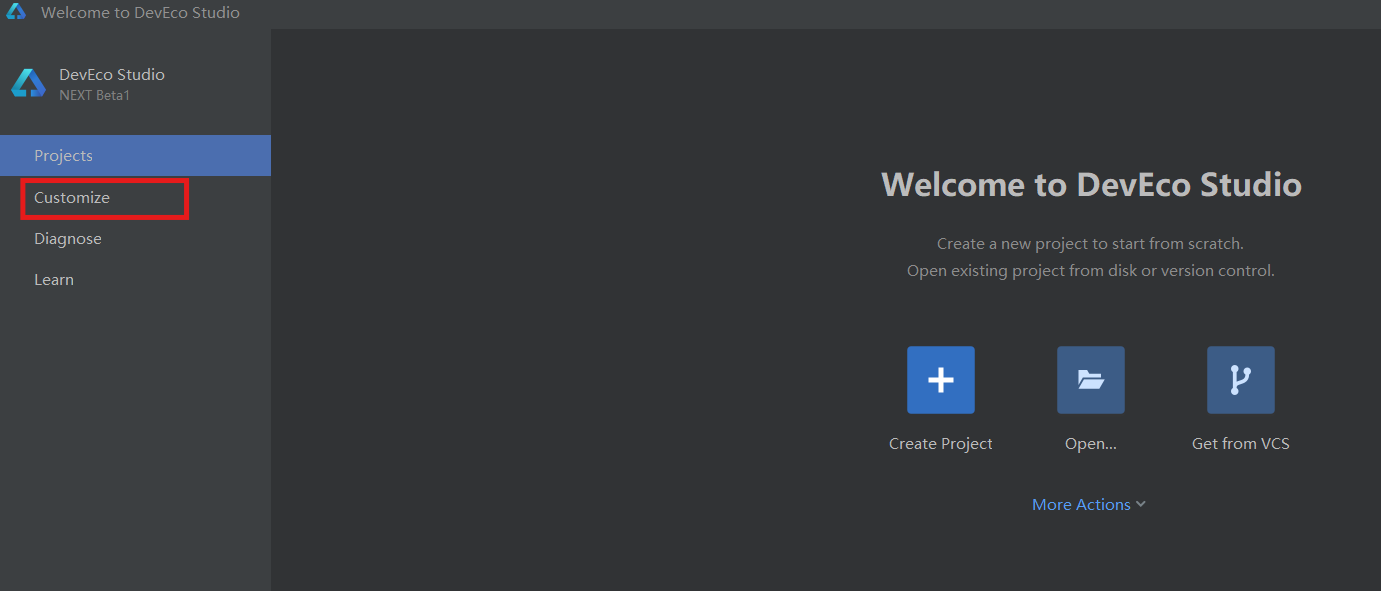


## 配置中文环境

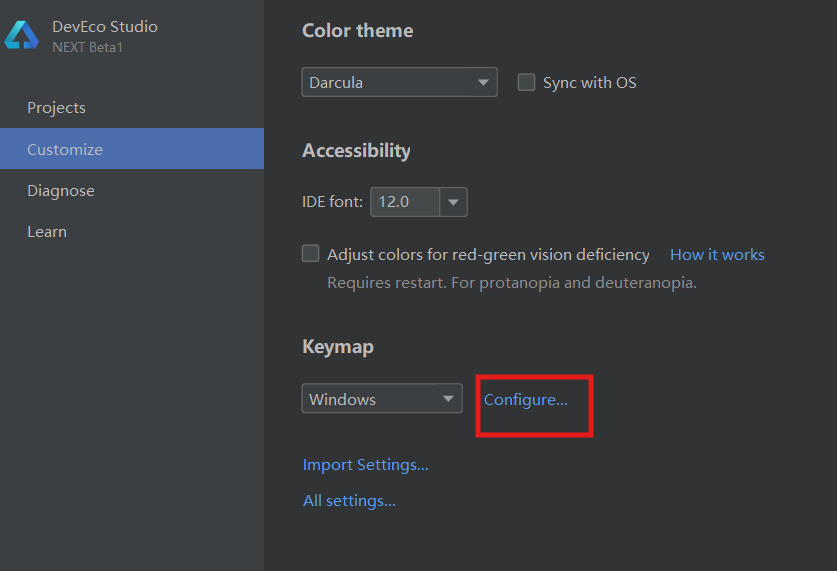
双击devecostudio64.exe，打开IDE，选择无配置文件进入，配置中文环境。



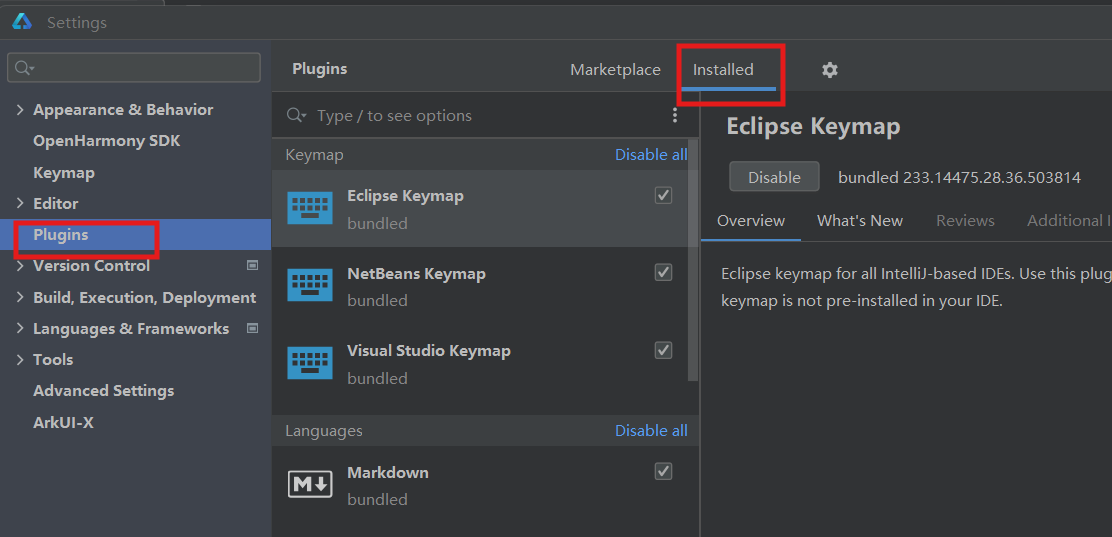
1. 选择Customize



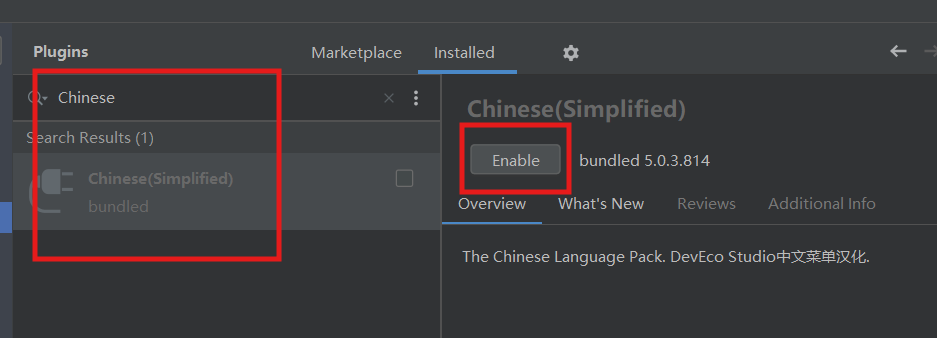
1. 选择configure

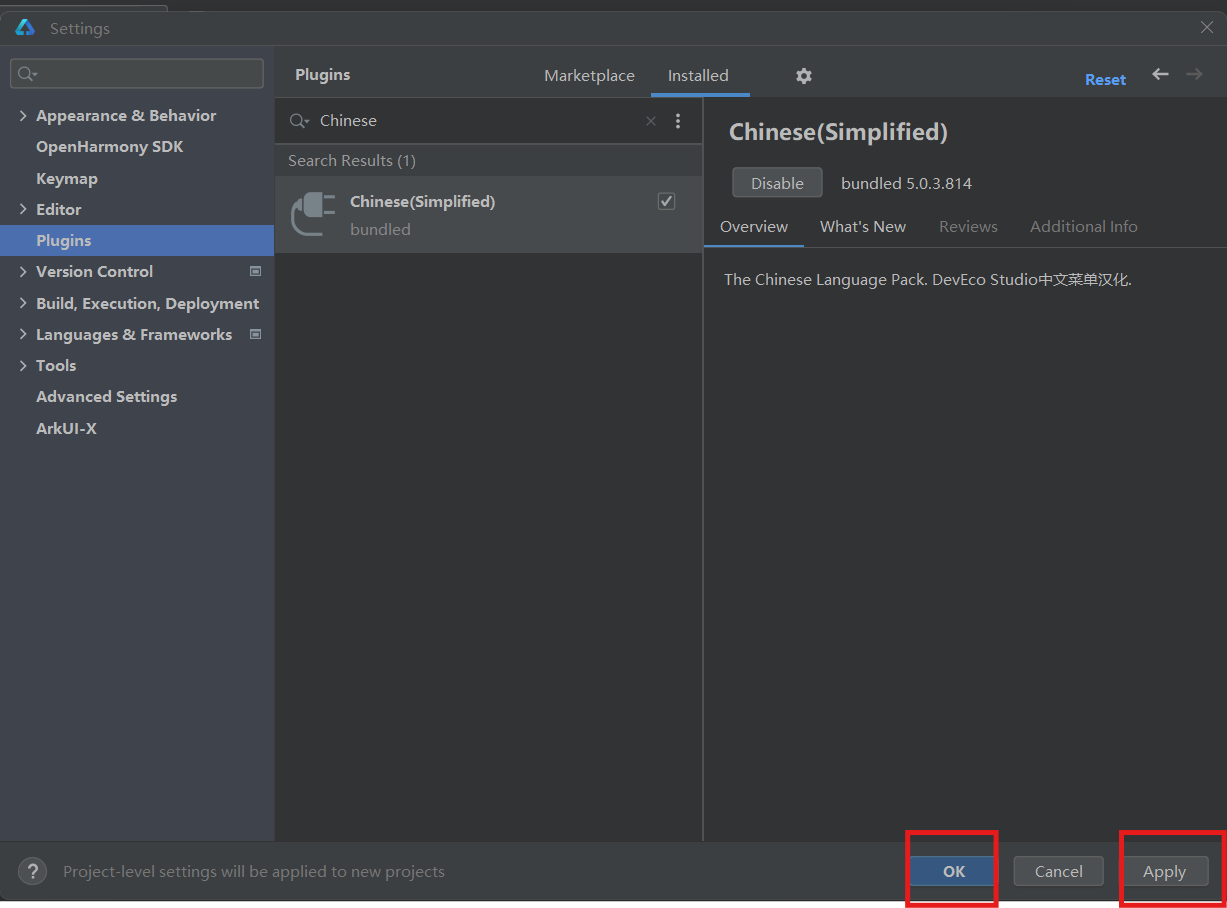


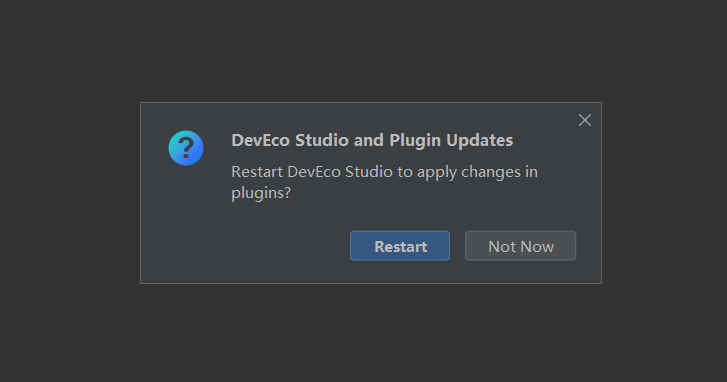
1. 选择plugins，然后选择installed



1. 搜索Chinese，选择插件enable，点击Apply，点击OK，重启客户端

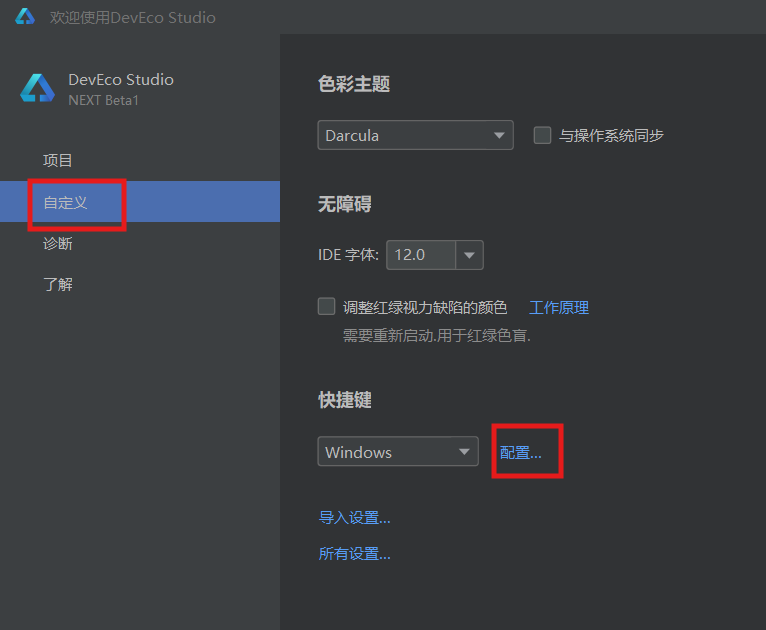






## 安装SDK

1. 点击自定义，选择配置：



1. 选择OpenHarmony SDK：



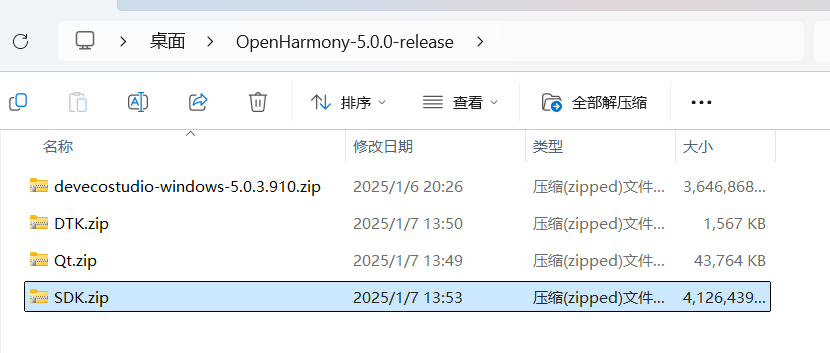
1. 点击编辑，选择之前新建的SDK，空目录：

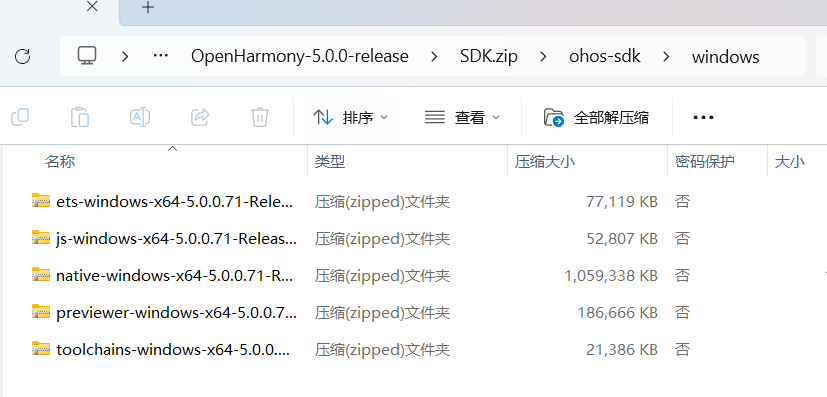


1. 先下载API 10，待API 10下载完成，找到刚选择的SDK目录，新建12目录：

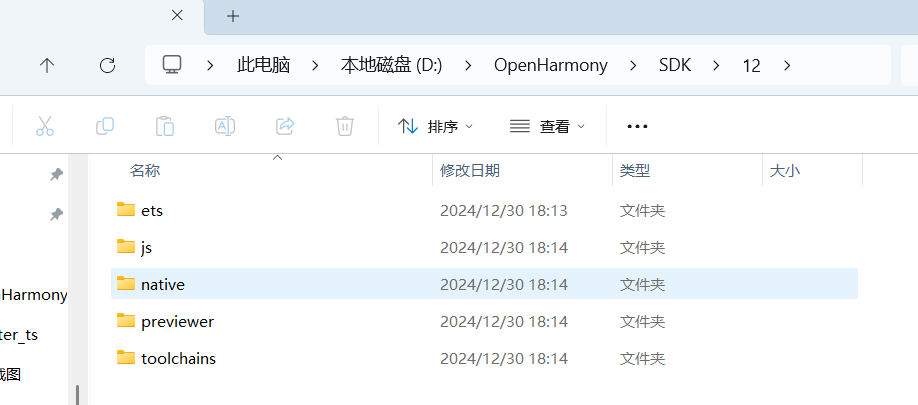


1. 将SDK(OpenHarmony-5.0.0-release\SDK.zip)解压, 取windows目录中的压缩包解压到D:\OpenHarmony\SDK\12目录中：

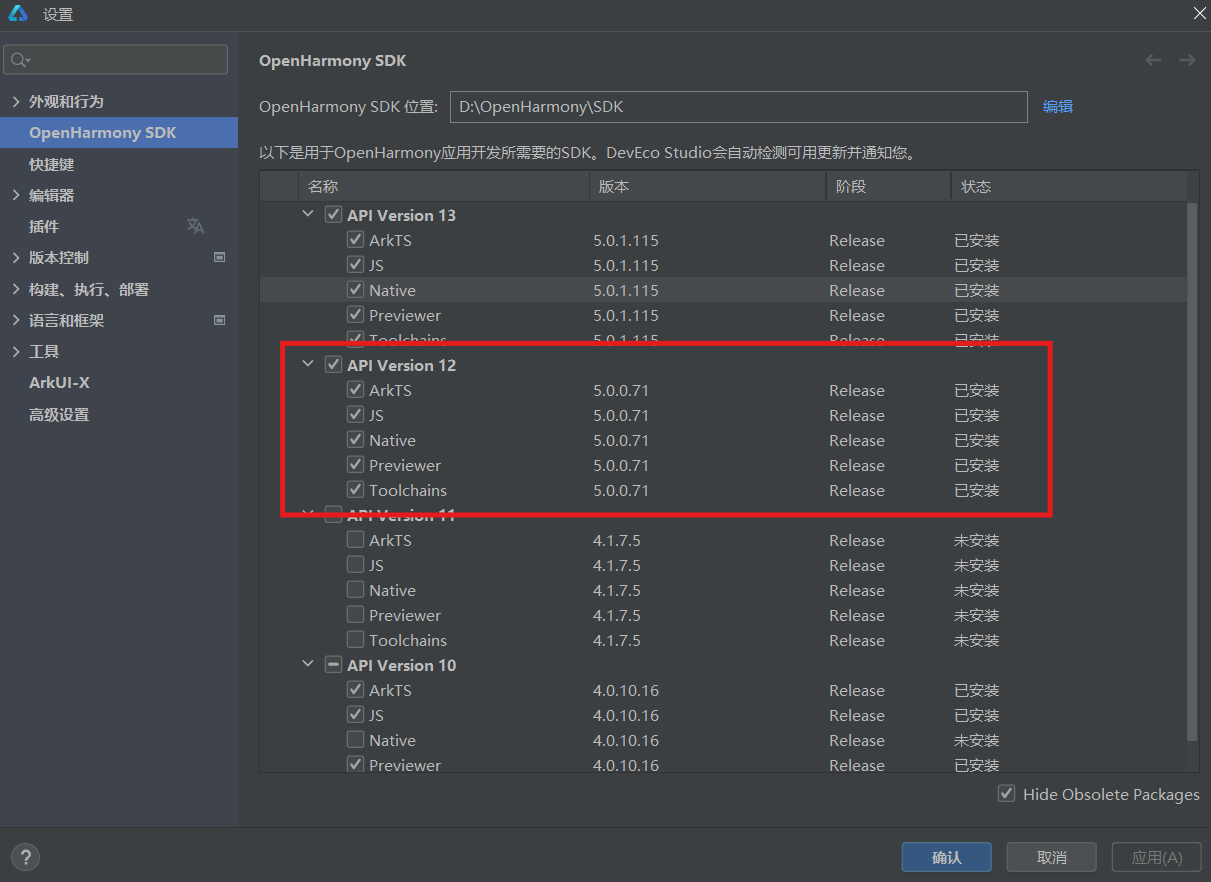




最后解压完目录结构如下：



1. 配置后IDE中会有如图API 10和12：



1. 点击确认，鸿蒙loongarch64 SDK配置完成。

## 配置ArtTS开发环境

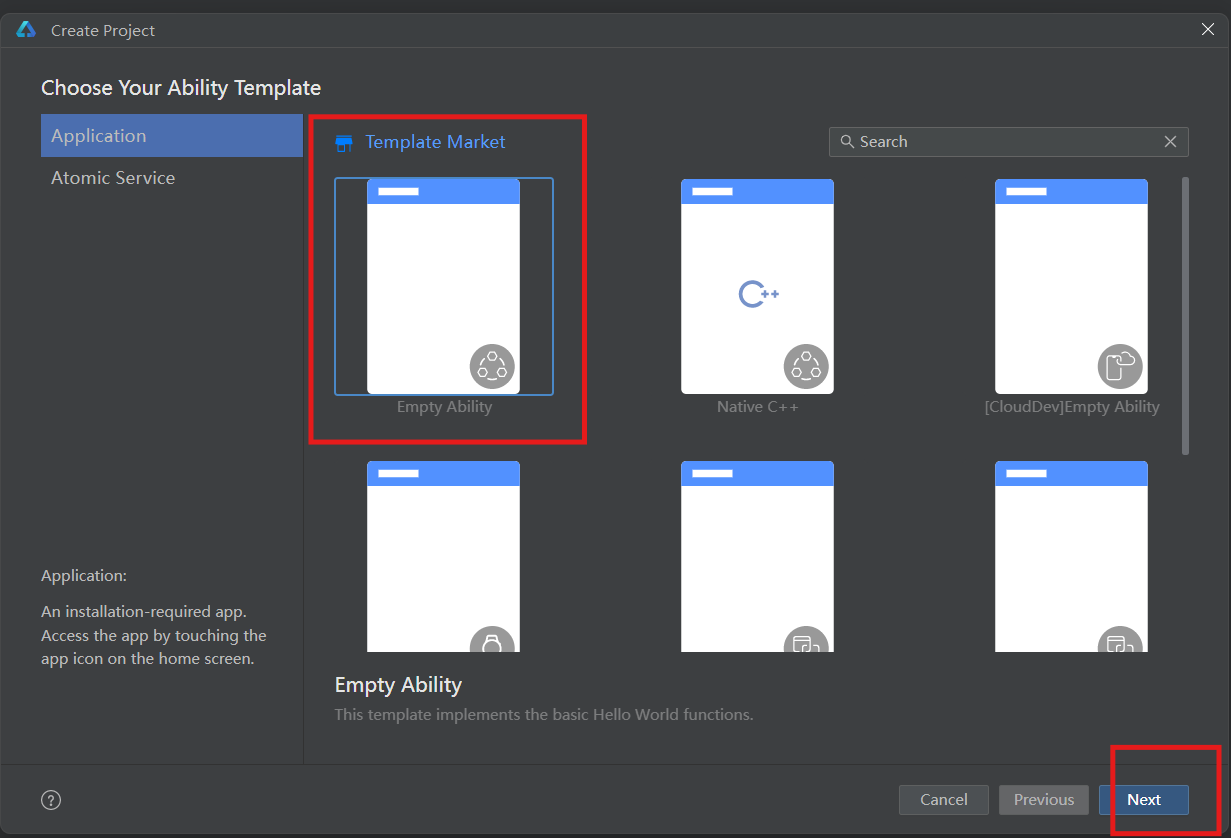
### 创建ArtTS项目

参考文档：[鸿蒙开发快速入门](https://developer.huawei.com/consumer/cn/doc/harmonyos-guides-V5/start-overview-V5)

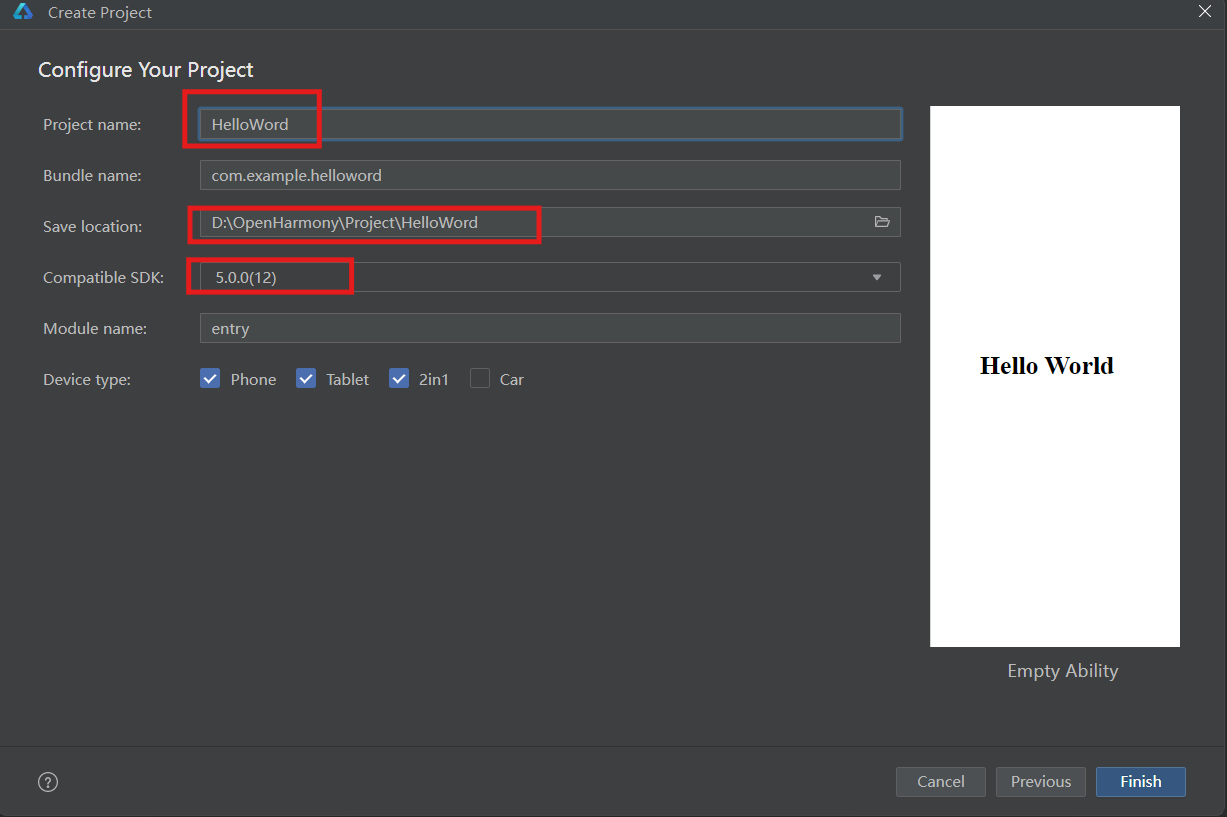
1. 创建ArtTS项目：



1. 选择Empty Ability，选择next：

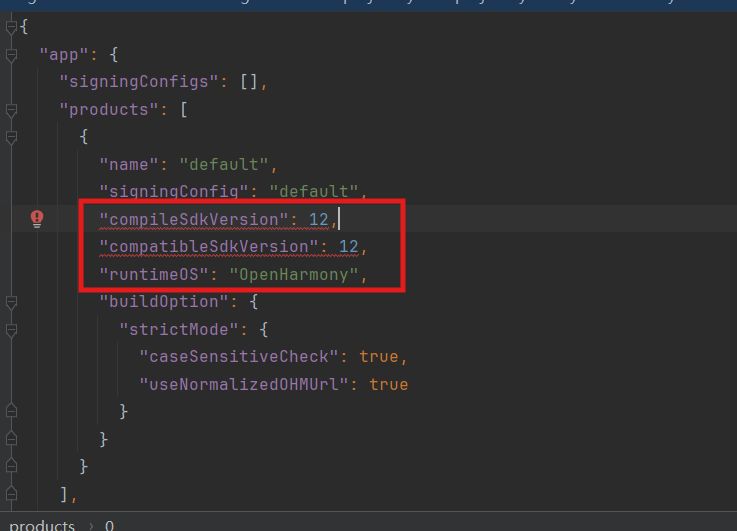


1. 进入项目配置界面，项目存储在之前新建的D:\OpenHarmony\Project目录下：

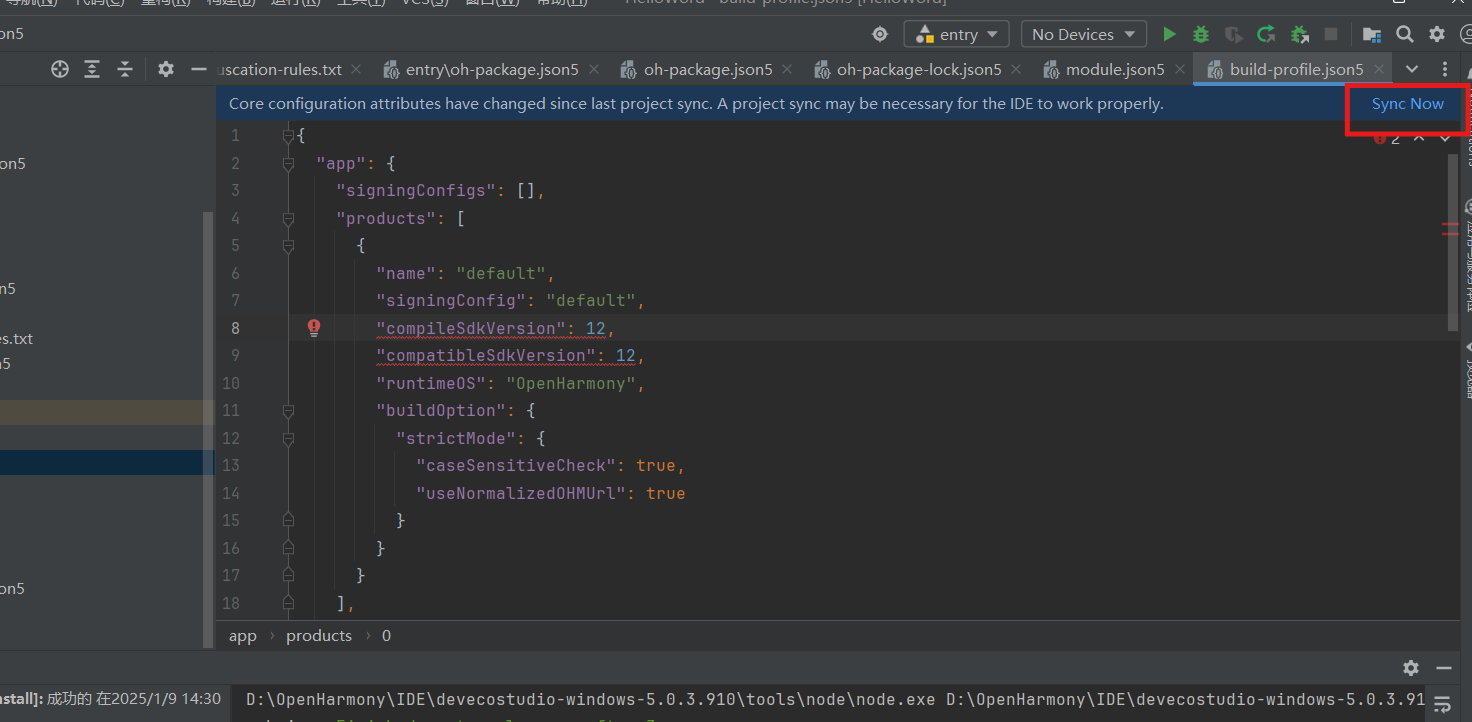


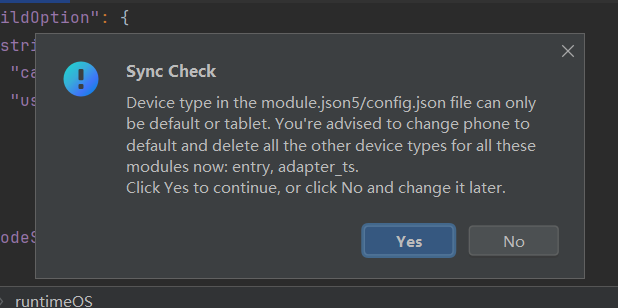
### 修改项目配置

修改D:\OpenHarmony\Project\HelloWord\build-profile.json5如下图：



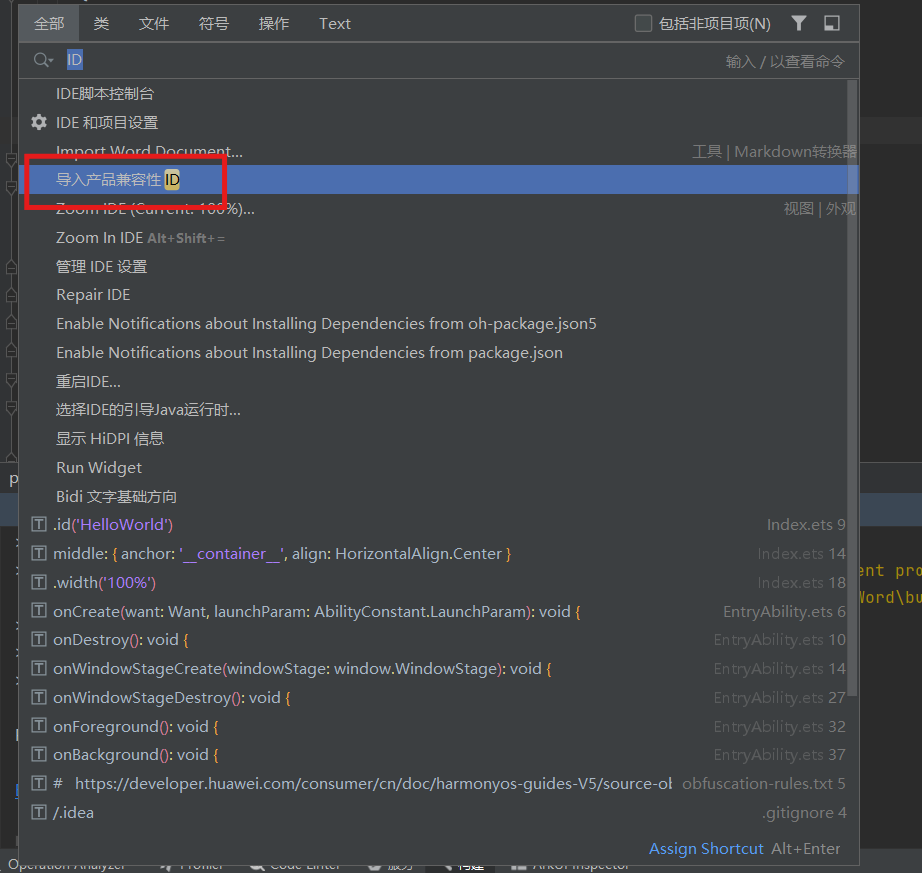
点击Sync Now，然后点击Yes，配置完成：



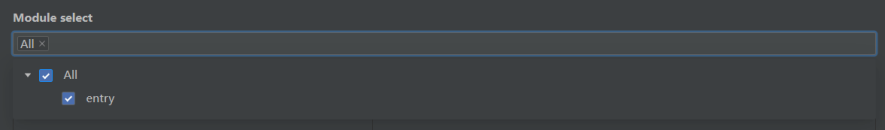


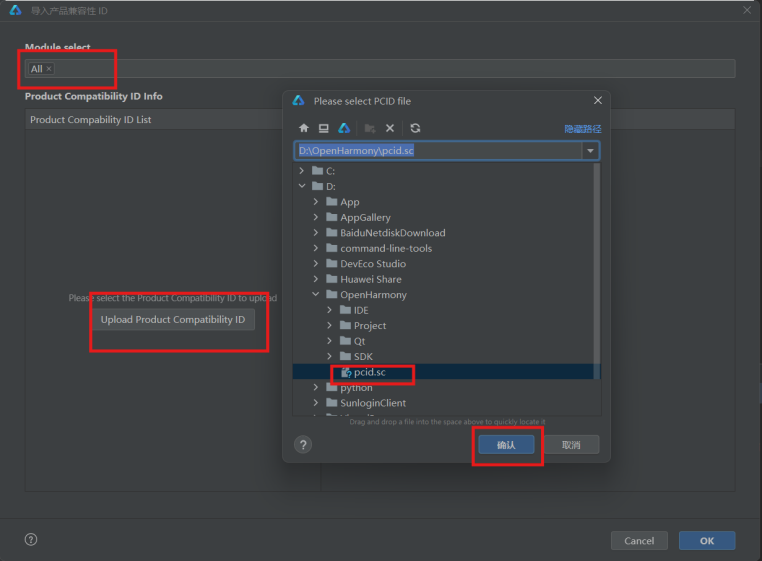
### 导入产品兼容性ID

在DevEco Studio界面双击shift，出现全局搜索框，搜索“导入产品兼容性ID”：

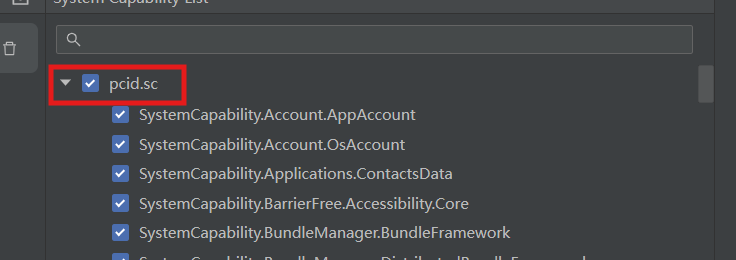


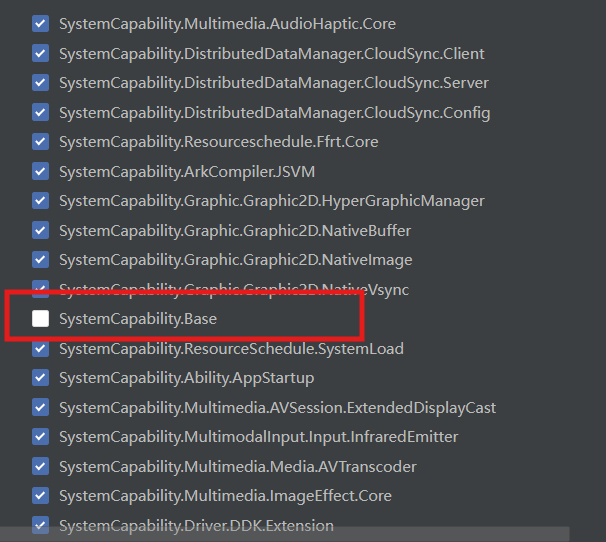
模块选择All，将OpenHarmony-5.0.0-release\pcid.sc保存到D:\OpenHarmony\pcid.sc，点击“Upload Product Compatibility ID”, 选择pcid.sc，点击确认：





展开pcid.sc，取消“SystemCapability.Base”的勾选，点击OK：





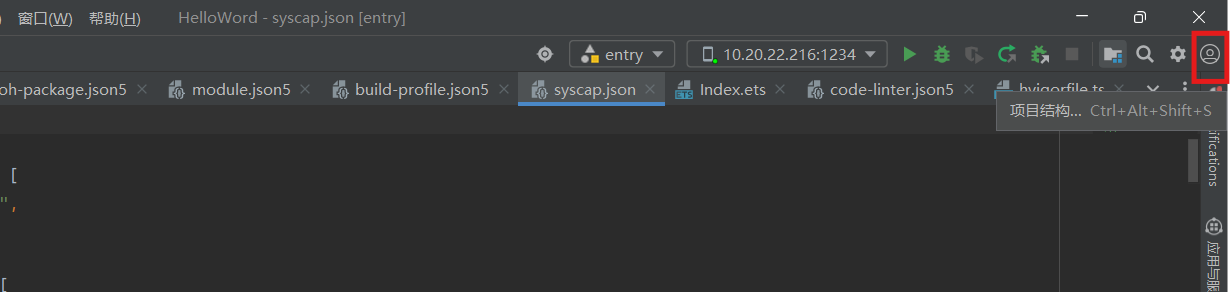
### 编写页面

工程同步完成后，D:\OpenHarmony\Project\HelloWord项目下，点击“entry > src > main > ets > pages”，打开“Index.ets”文件，进行页面的编写。

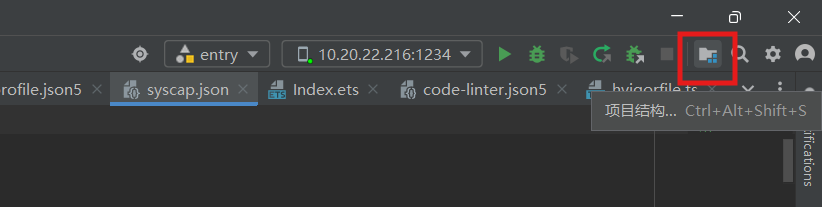
# 源码工程（应用级）加签

因鸿蒙机制要求，源码工程需配置签名。按如下步骤操作：

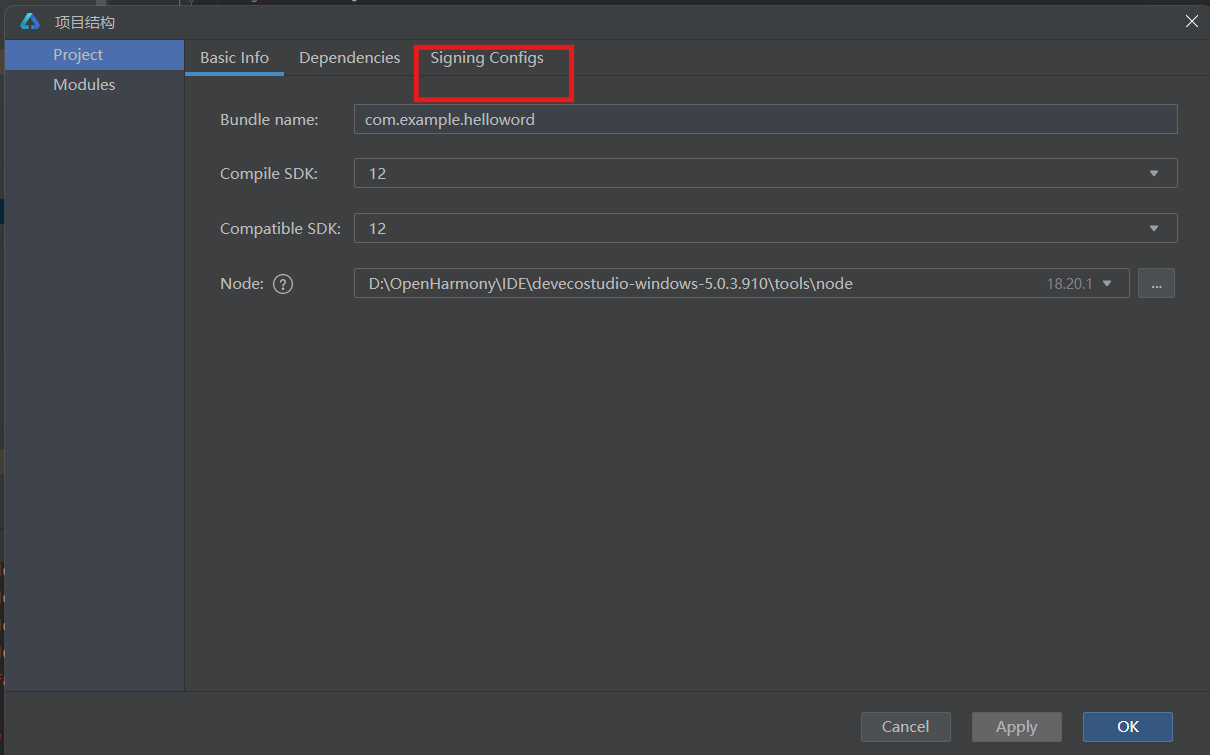
1. 登录华为账号：如下图，点击登录，如果没有账号，需要先注册；



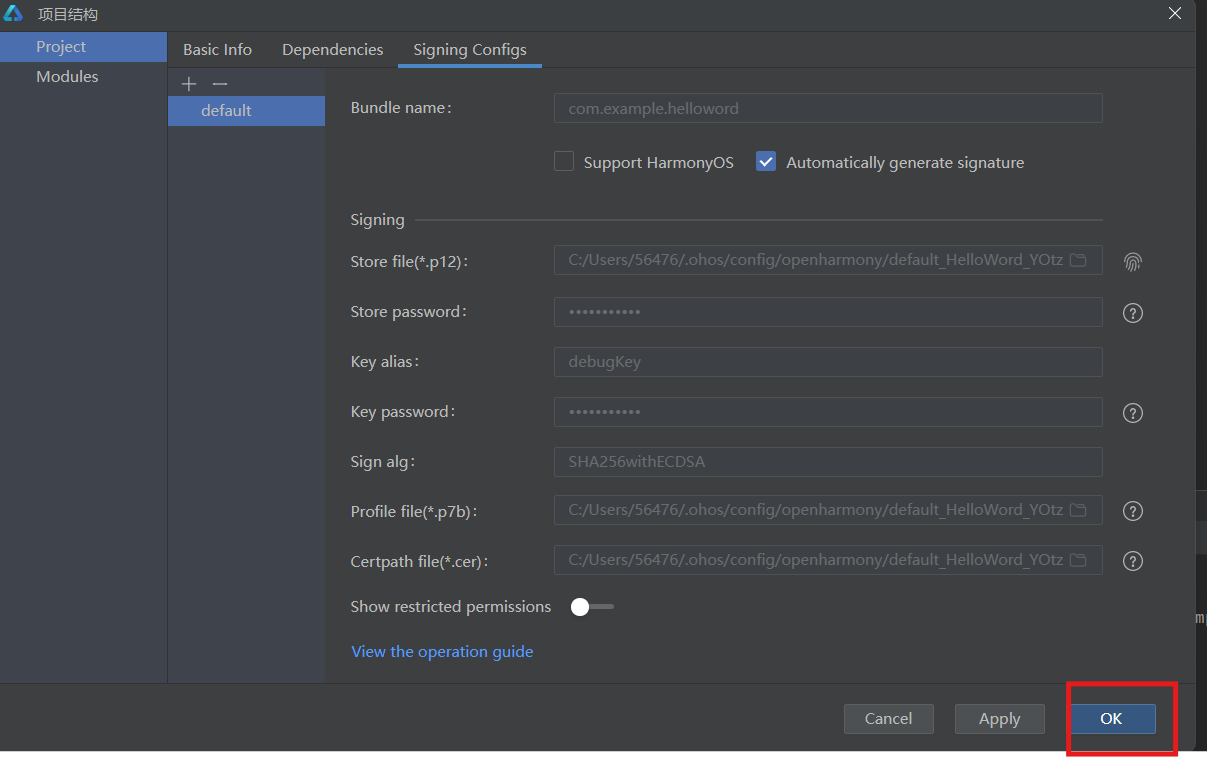
1. 点击”项目结构”：



1. 选择”Signing Configs”：



1. 等待页面加载完成后，点击OK加签。



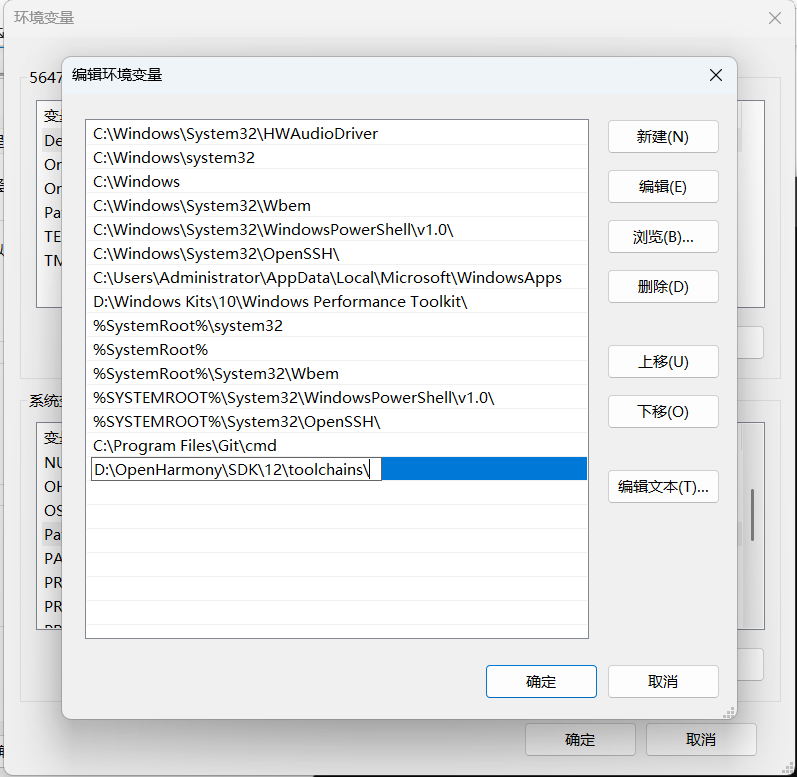
# 应用上传

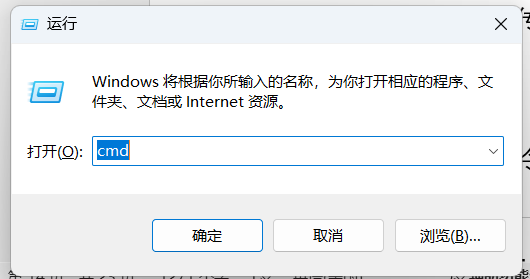
## hdc连接设备

查看3A6000电脑的IP，需要3A6000电脑与开发的Windows再同一网络下。

ChengHongOS上查看IP的方式可以通过“设置”=>“网络”=>“网络详情”中查看，连接命令如下：

如果将hdc配置到Windows的环境变量后，通过Win+R键，输入cmd，打开终端D:\OpenHarmony\SDK\12\toolchains, hdc程序再这个位置。

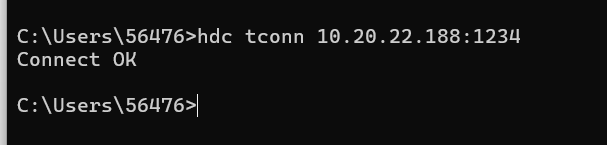
****



输入（例IP为10.20.22.188，1234为默认端口号）：

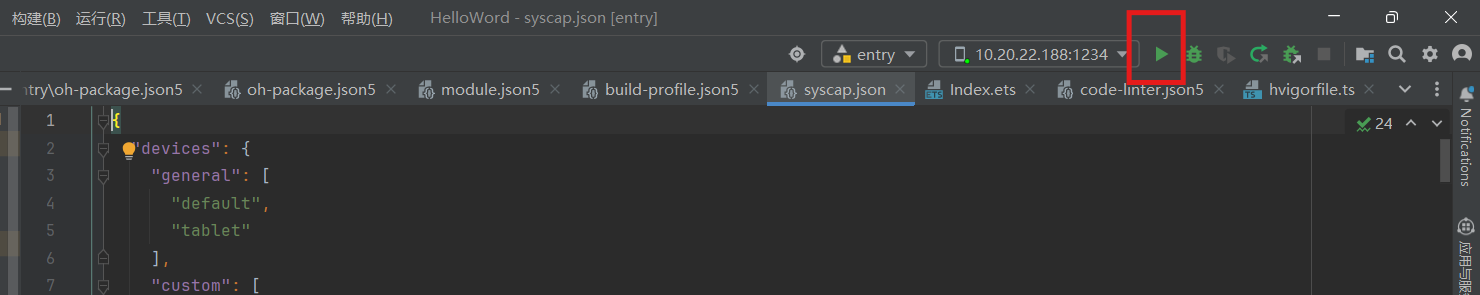
hdc tconn 10.20.22.188:1234

连接成功显示如下：



## IDE上传和运行

点击开始按钮，上传并运行，实际也是hdc命令安装：



hdc命令行上传并运行：

# 强制关闭之前运行的应用

hdc shell aa force-stop com.example.helloword

# 建一个临时存放的目录

hdc shell mkdir data/local/tmp/3abdb2db43d34cffab35cc500ade65e5

# 将生成的加签后的hap包传到3A6000机器上

hdc file send D:\OpenHarmony\Project\HelloWord\entry\build\default\outputs\default\entry-default-signed.hap "data/local/tmp/3abdb2db43d34cffab35cc500ade65e5"

# 安装应用

hdc shell bm install -p data/local/tmp/3abdb2db43d34cffab35cc500ade65e5

# 删除之前建的临时目录和安装包

hdc shell rm -rf data/local/tmp/3abdb2db43d34cffab35cc500ade65e5

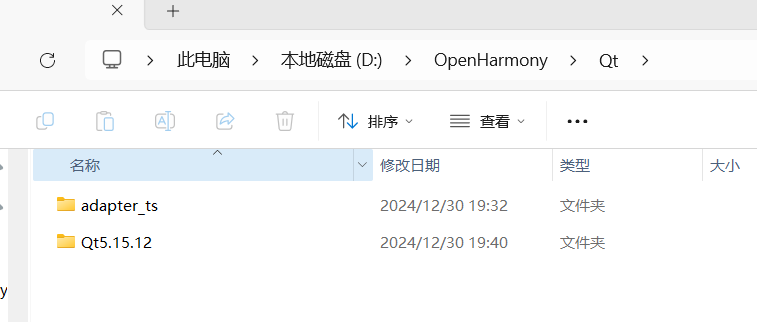
# 启动应用

hdc shell aa start -a EntryAbility -b com.example.helloword

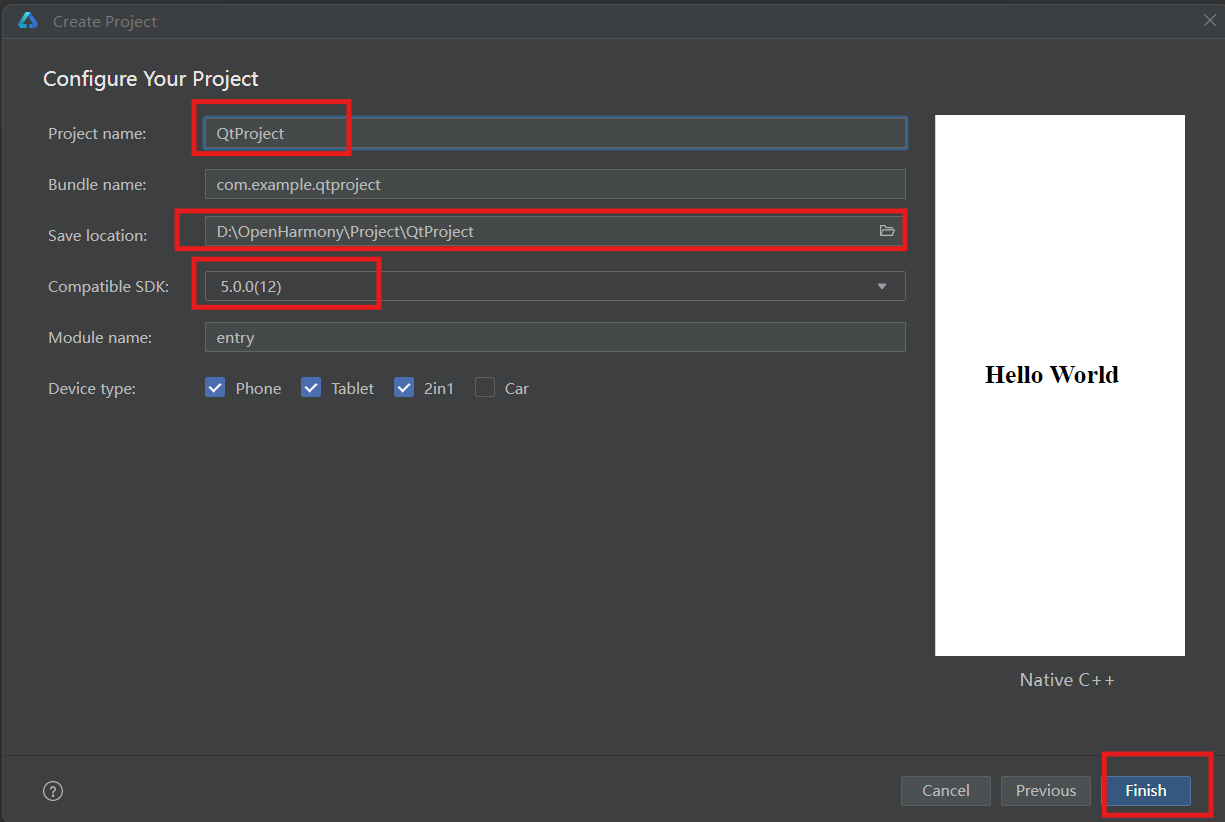
# 配置Qt开发环境与开发

## 配置Qt开发环境

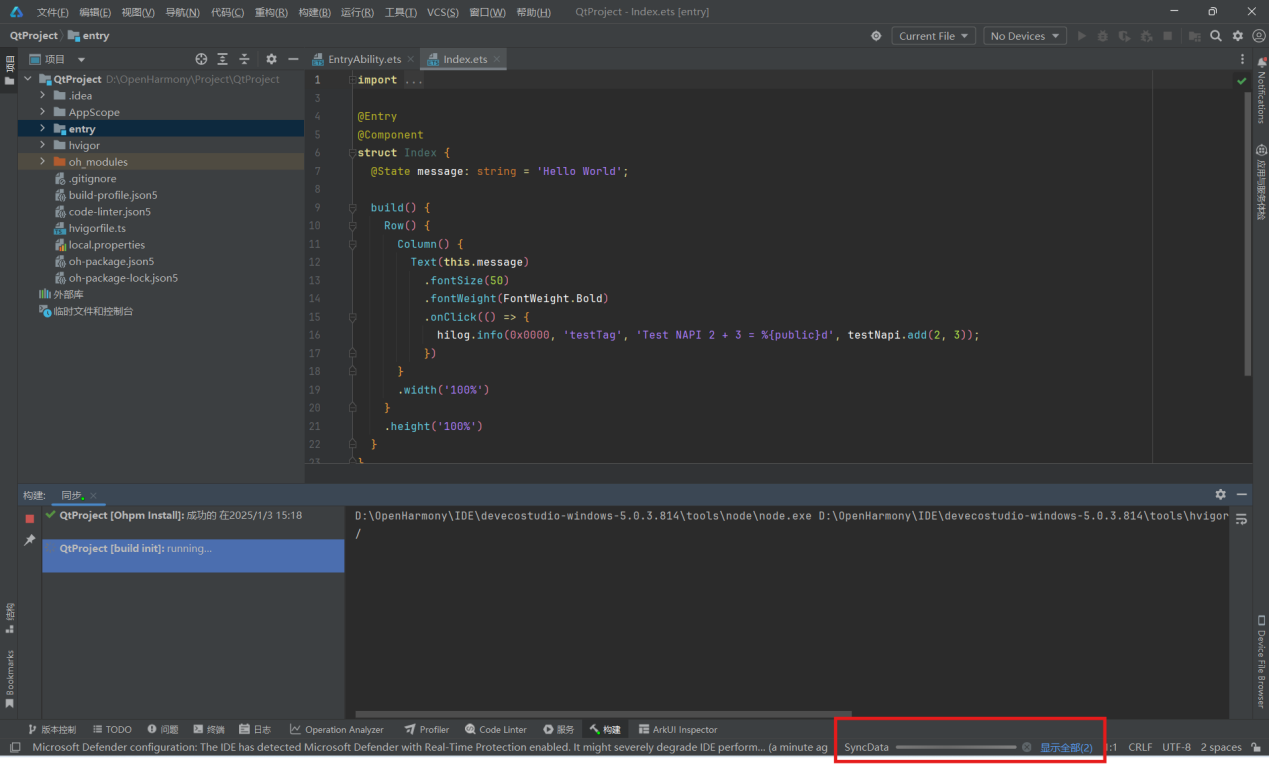
1. 解压OpenHarmony-5.0.0-release\Qt.zip到之前建的D:\OpenHarmony\Qt目录下：



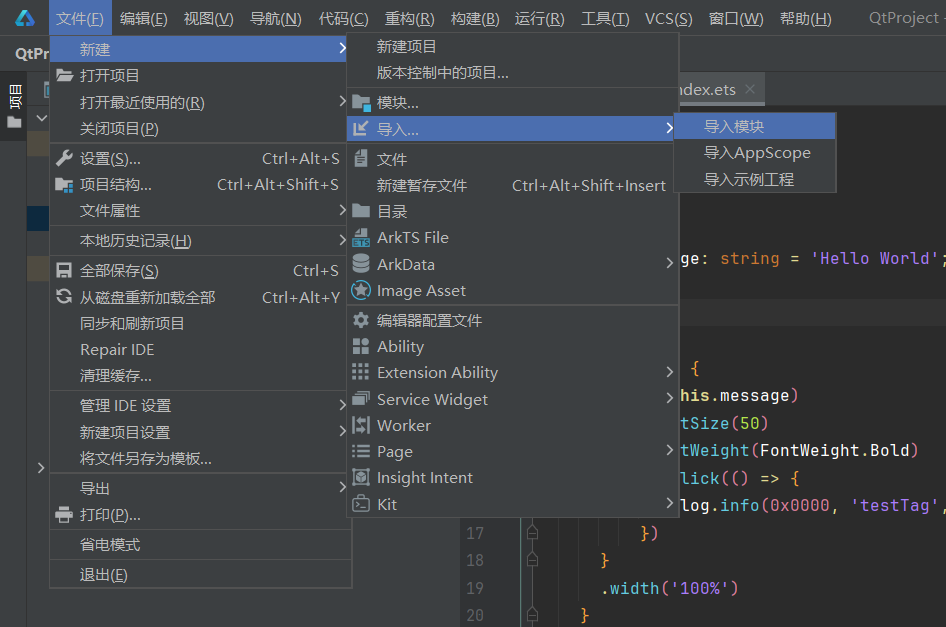
1. 新建项目，选择Native C++：
2. 选择之前的D:\OpenHarmony\Project目录存放，新建项目名称为QtProject，module名称为entry，点击finish：



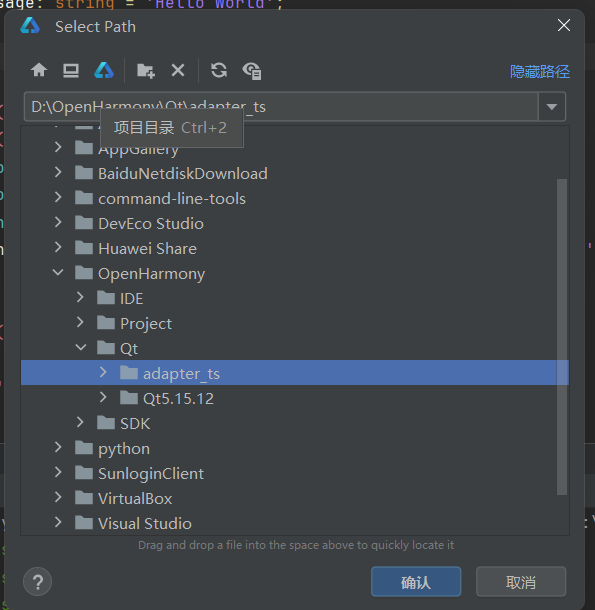
1. 新建项目等待初始化完成：

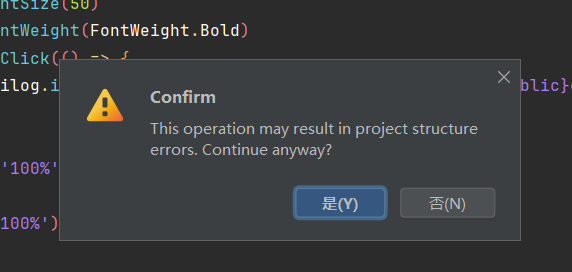


1. 导入adapter\_ts，修改adapter\_ts相关内容。导入模块：

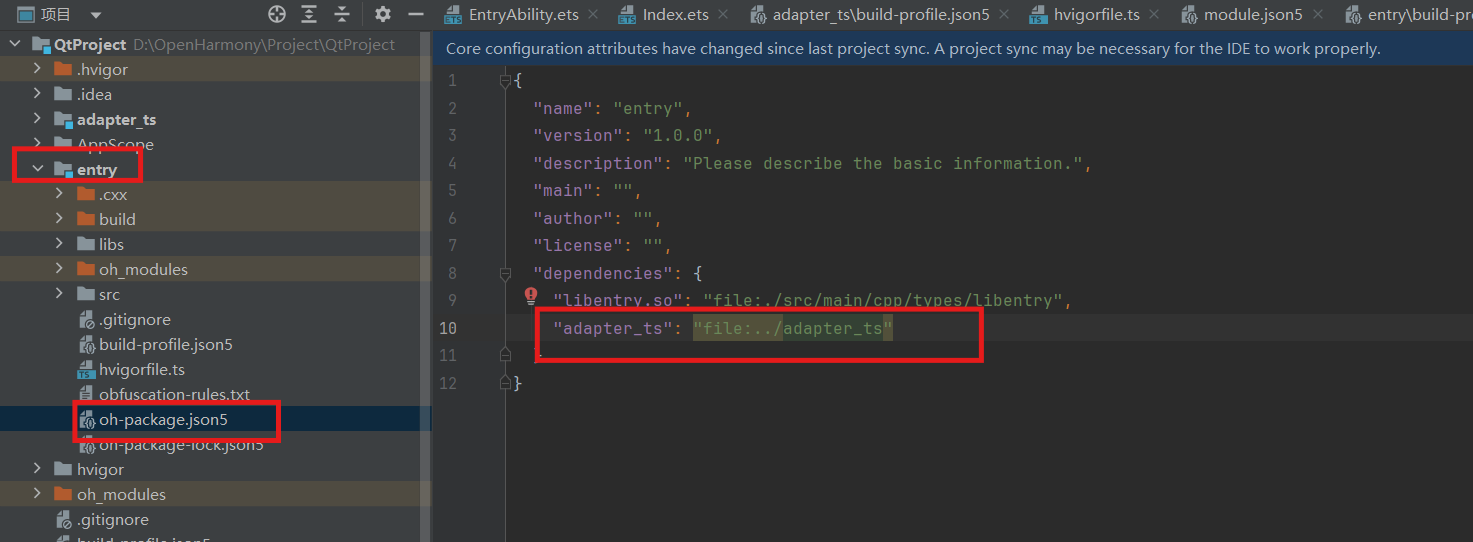


1. 选择adapter\_ts, 点击确认，弹出窗口再次确认窗口，点击是：

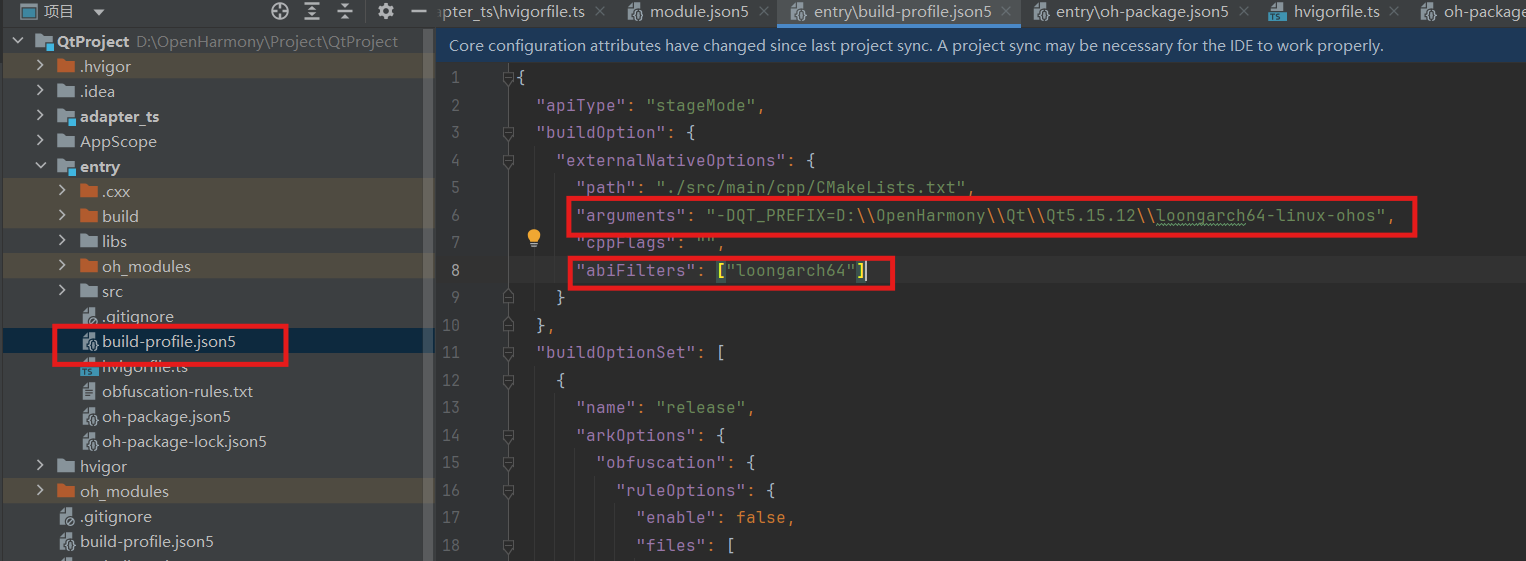




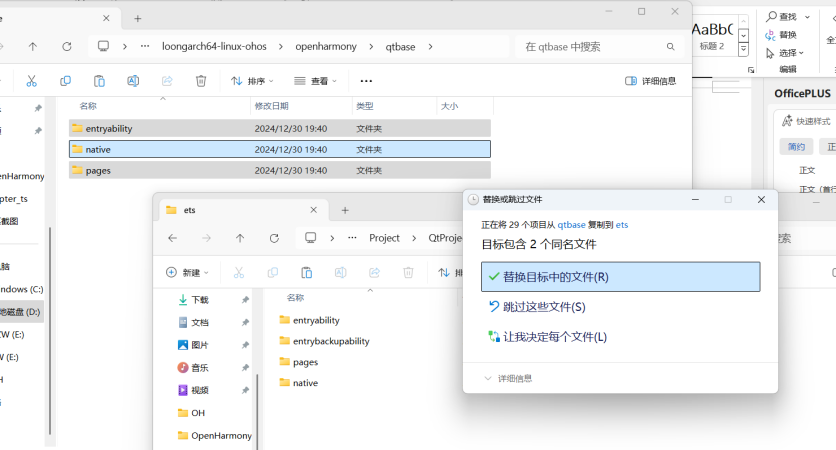
1. 修改QtProject/entry/oh-package.json5文件，添加dependencies: “adapter\_ts”: “file:../adapter\_ts”：

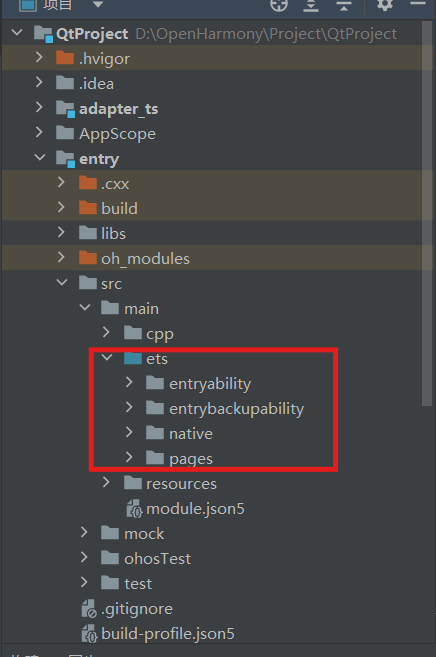


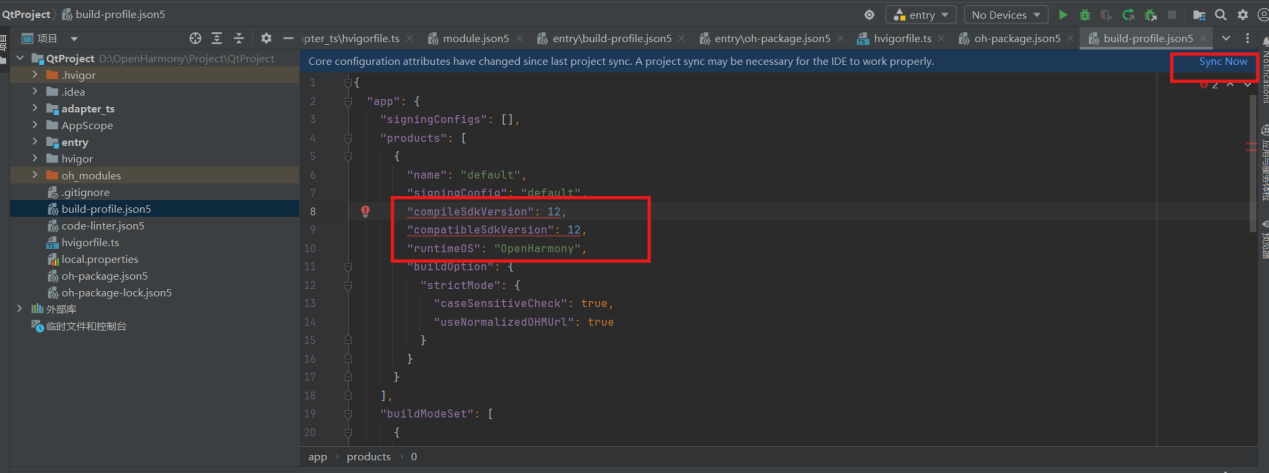
1. 修改QtProject/entry/build-profile.json5配置loongarch64和Qt环境。Qt环境配置在argruments参数中，abiFilters指定loongarch64：

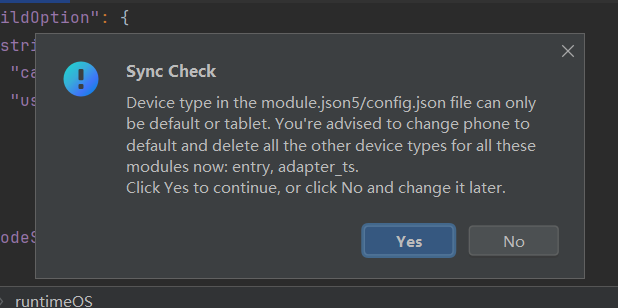
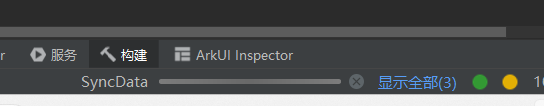


1. 替换JS接口文件，将D:\OpenHarmony\Qt\Qt5.15.12\loongarch64-linux-ohos\openharmony\qtbase目录下的文件替换到D:\OpenHarmony\Project\QtProject\entry\src\main\ets目录中：

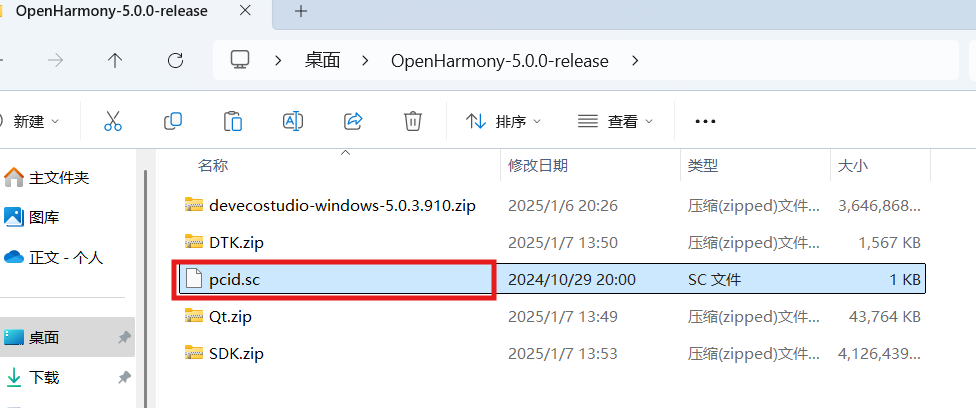


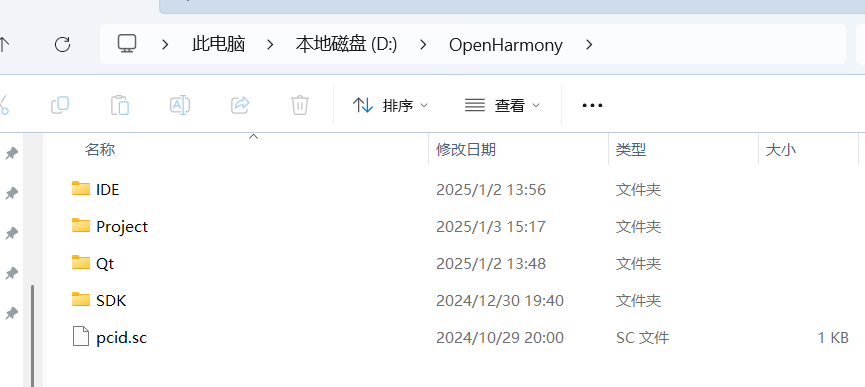


1. 配置SDK版本：QtProject\build-profile.json5中配置compileSdkVersion和compatibleSdkVersion版本为12，修改runtimeOS为OpenHarmony，点击右上角Sync Now，同步配置，点击yes，等待同步完成：

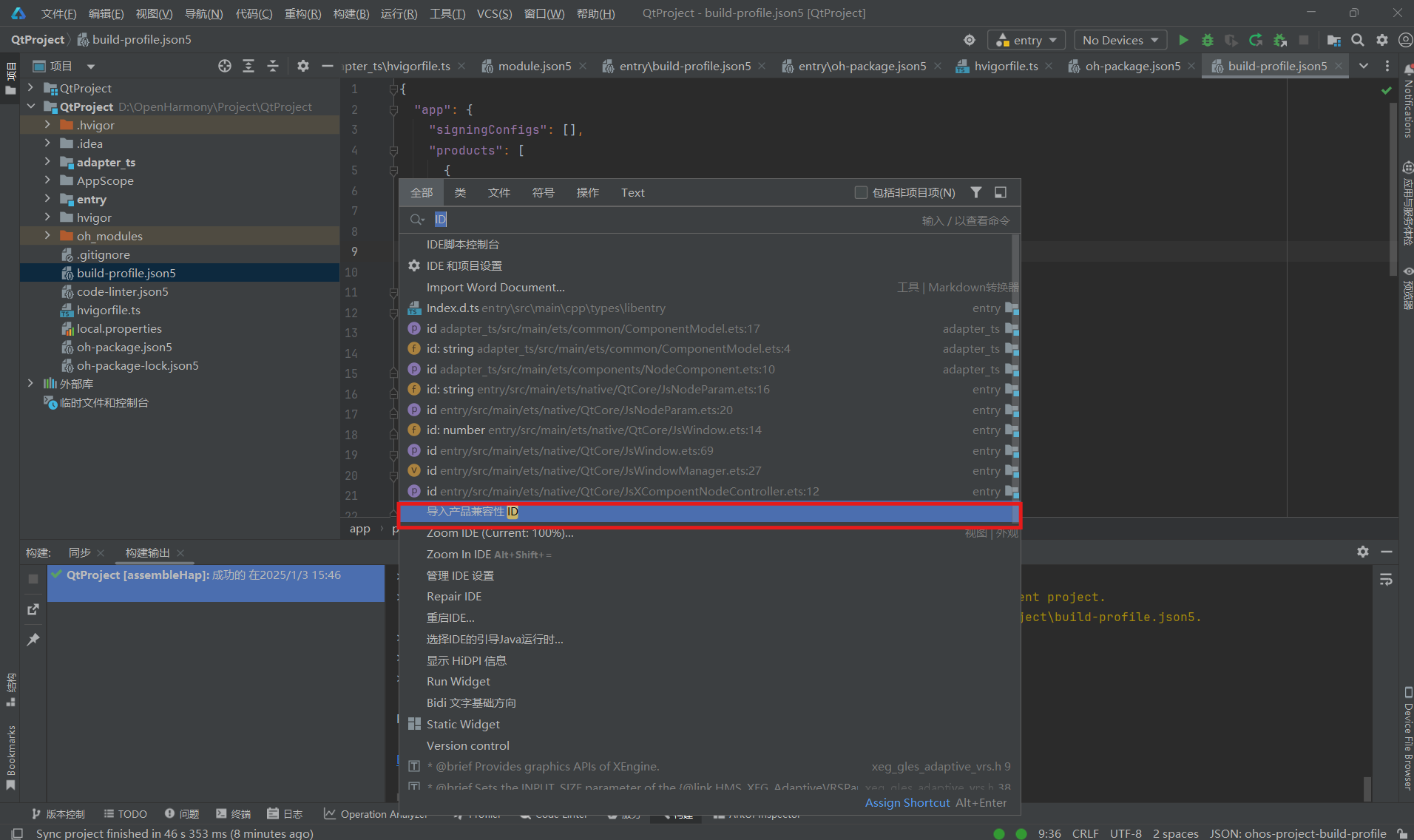
 

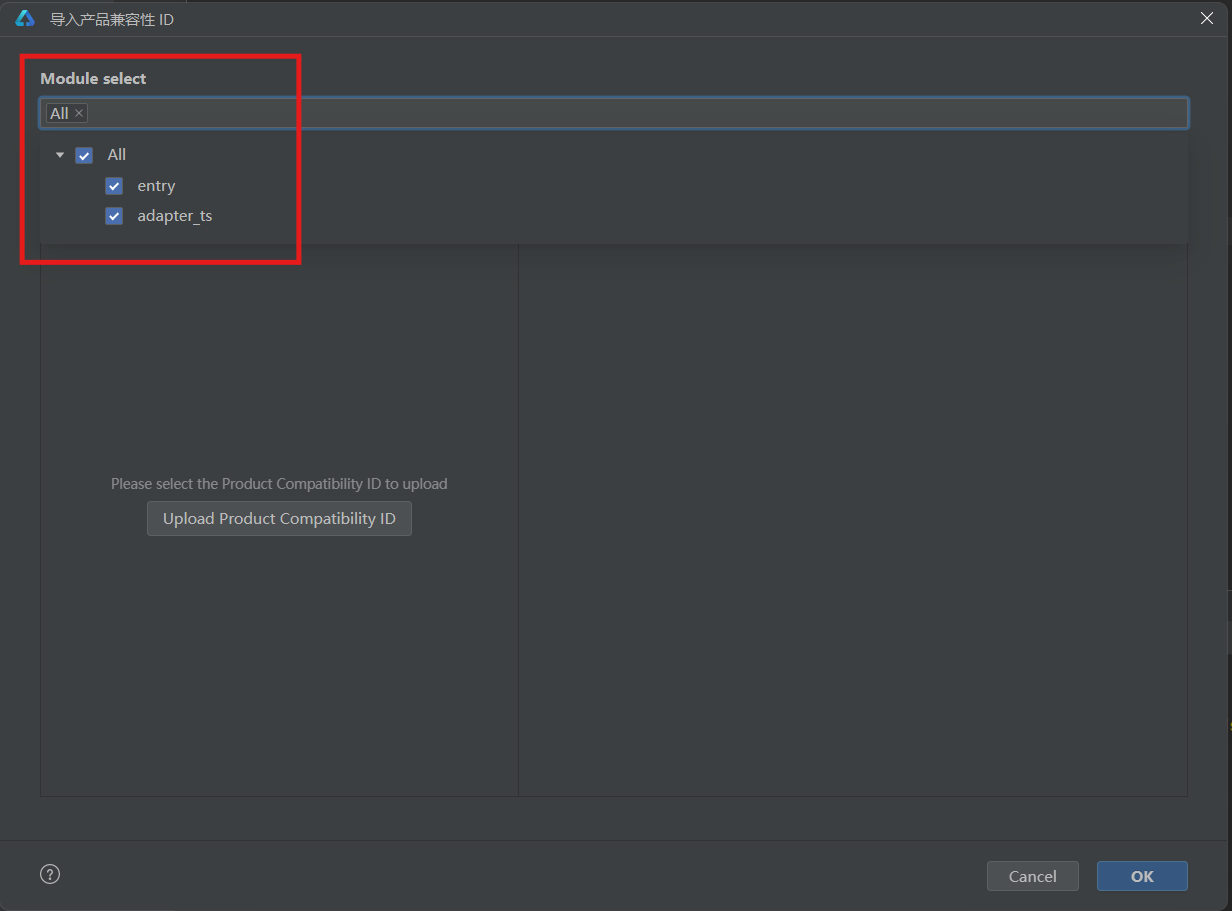
1. 导入pcid：将OpenHarmony-5.0.0-release\pcid.sc拷贝到D:\OpenHarmony\目录下：



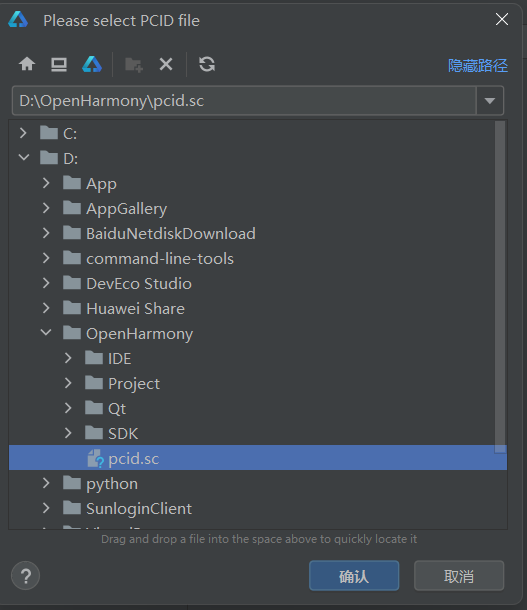


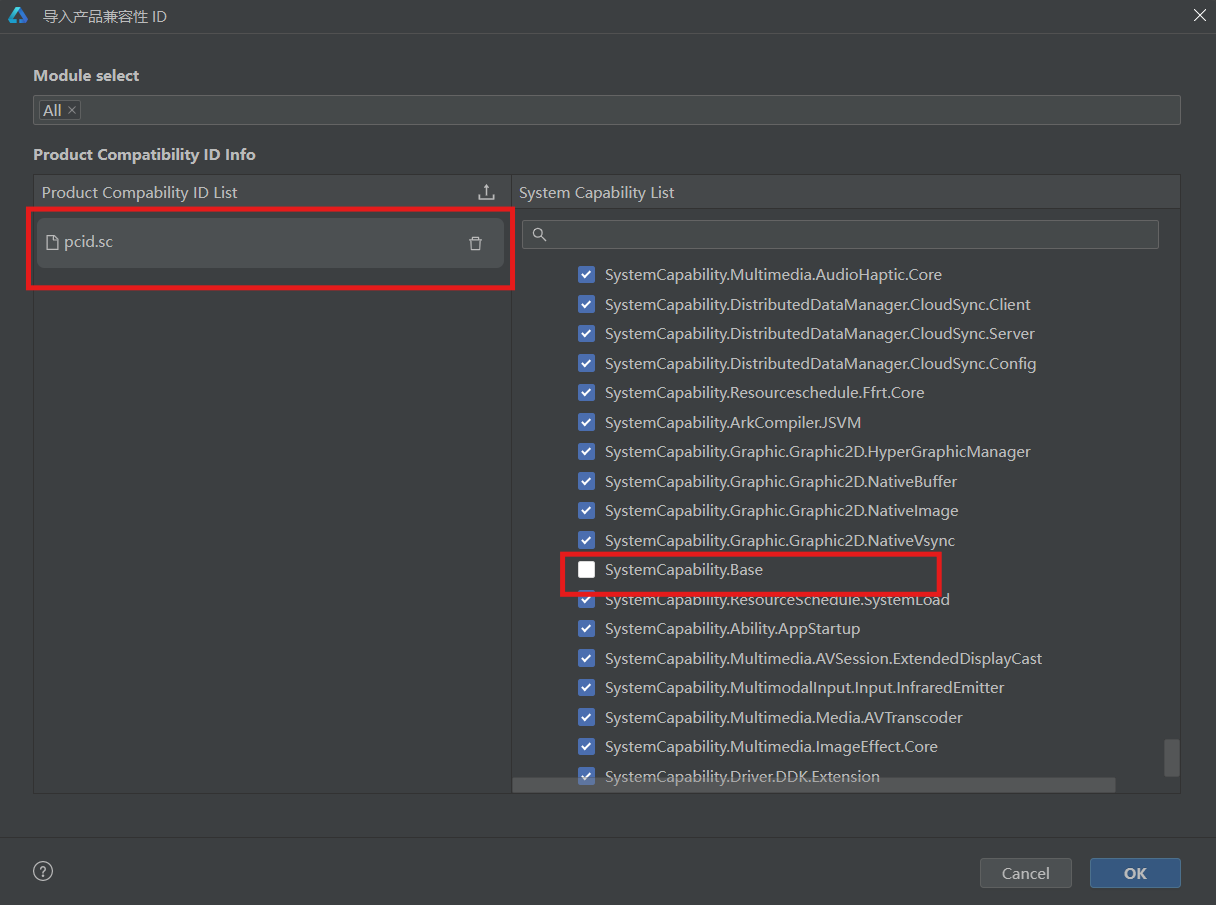
1. 在DevEco Studio界面下，双击shift，打开全局搜索，搜索“导入产品兼容性ID”，Module select选择All：

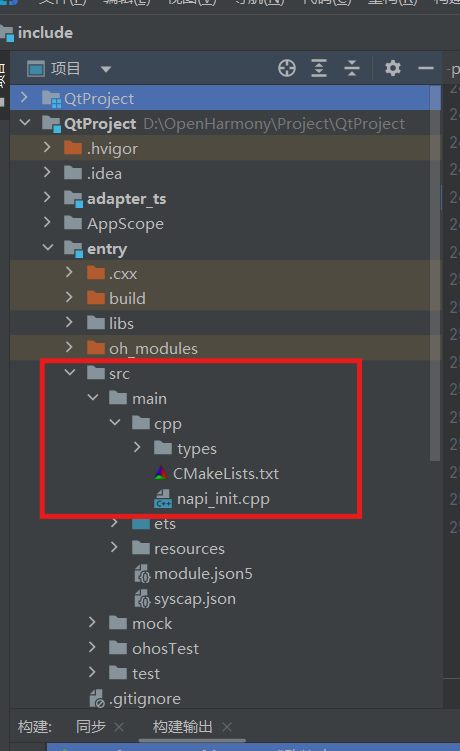




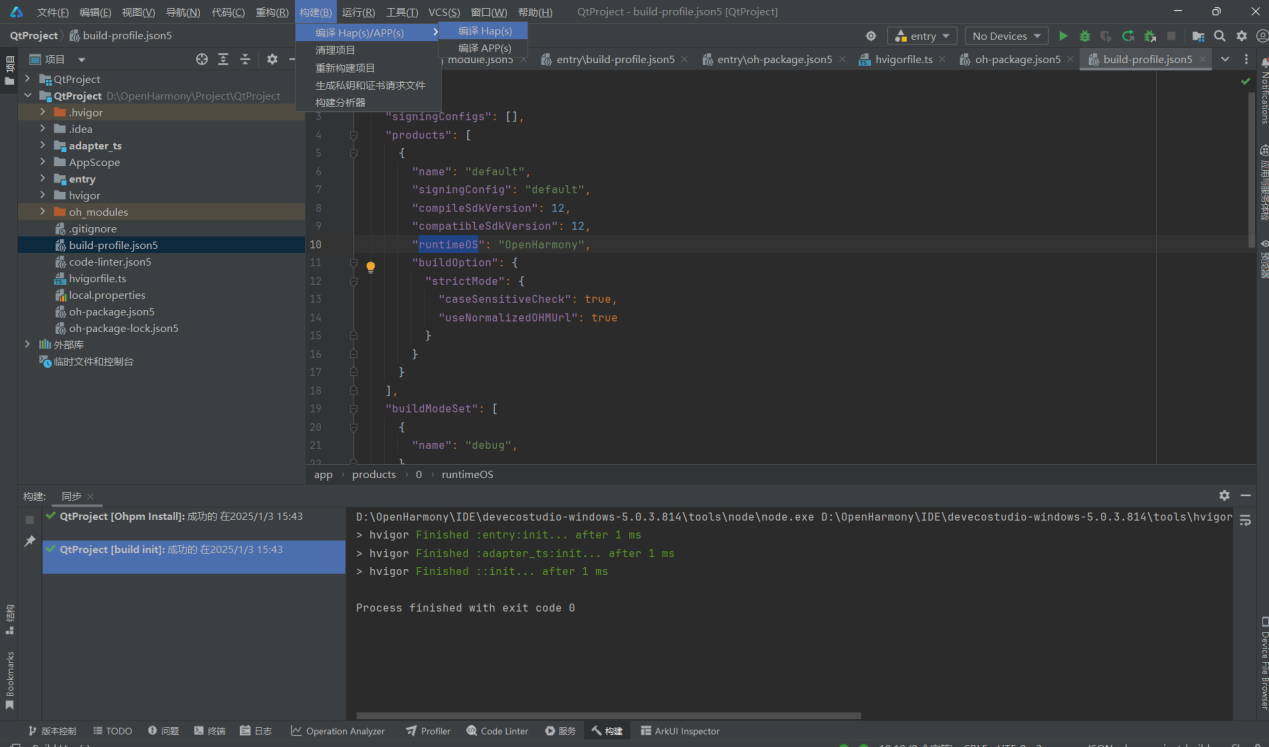
1. ID导入pcid.sc文件，取消勾选SystemCapability.Base，点击OK：







1. 编译项目：在顶部标题栏选择构建 => 编译Hap => 编译Hap(s)：



编译完成后，hap包在D:\OpenHarmony\Project\QtProject\entry\build\default\outputs\default目录下。

## 编写Qt项目

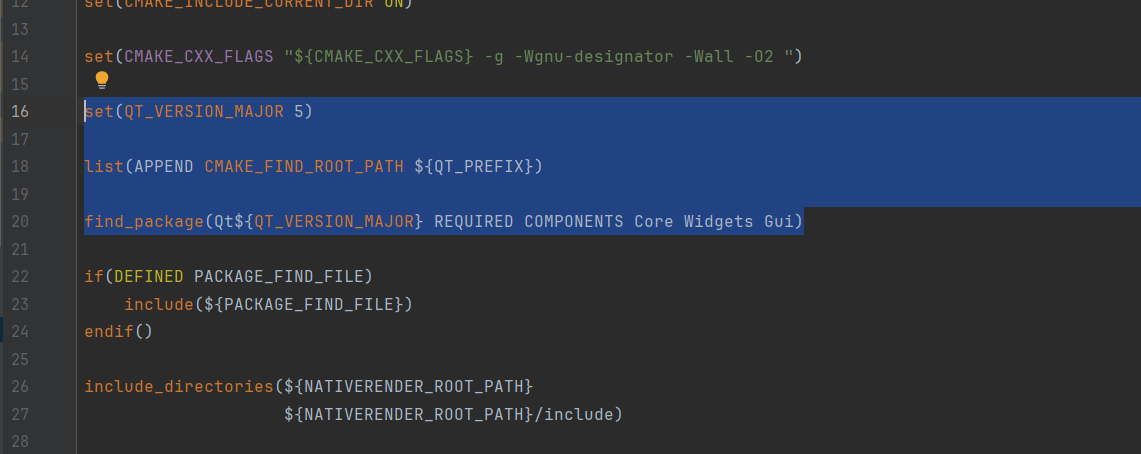
参考如下链接：

[OpenHarmony-sig/qt](https://gitee.com/openharmony-sig/qt/wikis/%E5%9F%BA%E4%BA%8EDevEco%E7%9A%84Qt%E5%B7%A5%E7%A8%8B%E9%85%8D%E7%BD%AE)

[OpenHarmony Qt源码地址](https://gitee.com/CplusCplus/qt5)

1. 修改CMakeLists.txt：

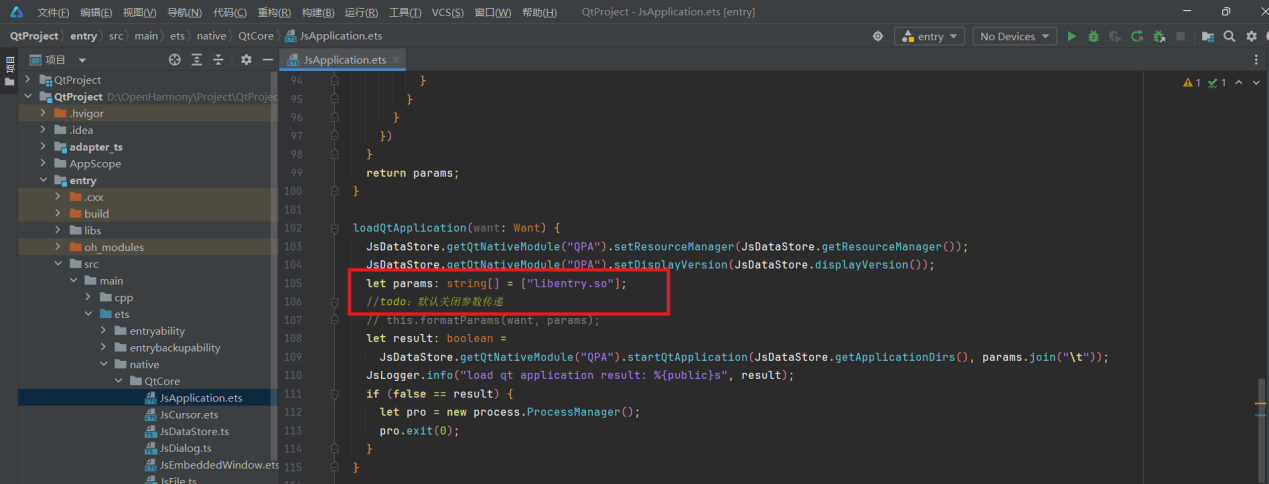
D:\OpenHarmony\Project\QtProject\ entry \src\main\cpp\CMakeLists.txt中需要在find\_package(Qt5 REQUIRED COMPONENTS Core Widgets Gui)前添加：list(APPEND CMAKE\_FIND\_ROOT\_PATH ${QT\_PREFIX})



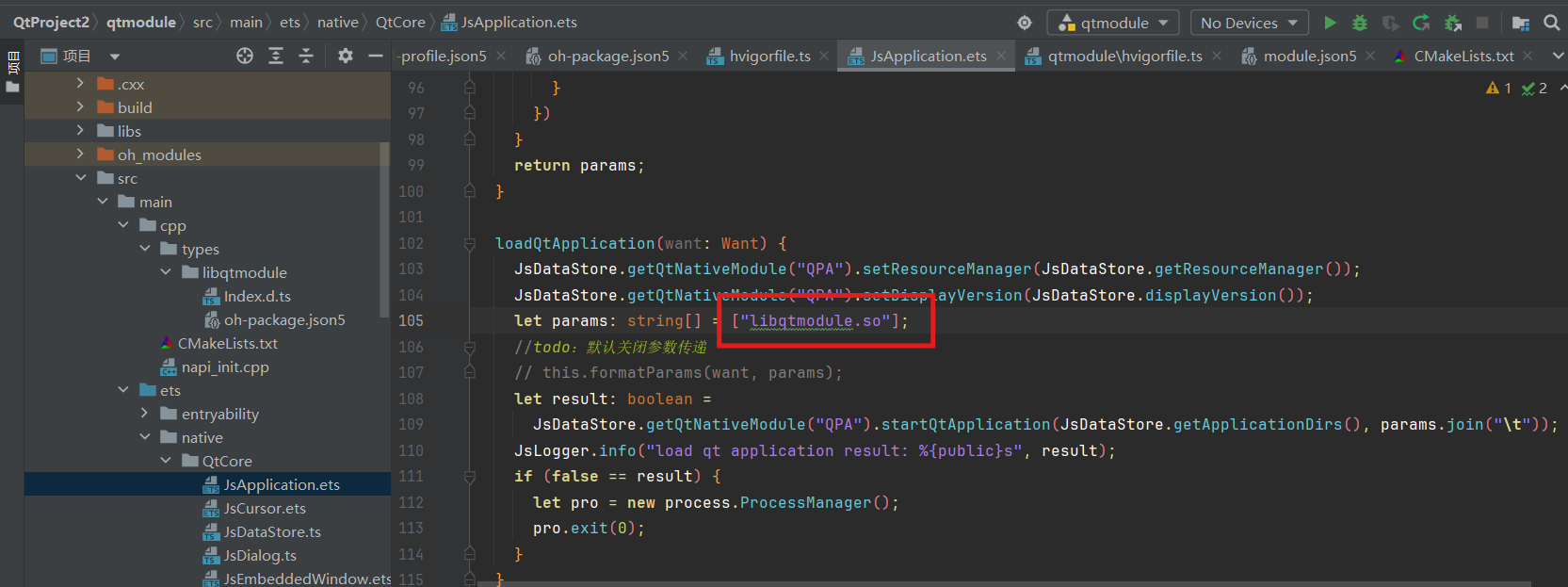
1. 修改加载的Qt程序：

修改QtProject\entry\src\main\ets\native\目录下的JsApplication.ts文件，将加载的Qt程序修改为cpp端生成的库文件，如下图所示：

如果新建项目时module时entry，这里就是libentry.so；

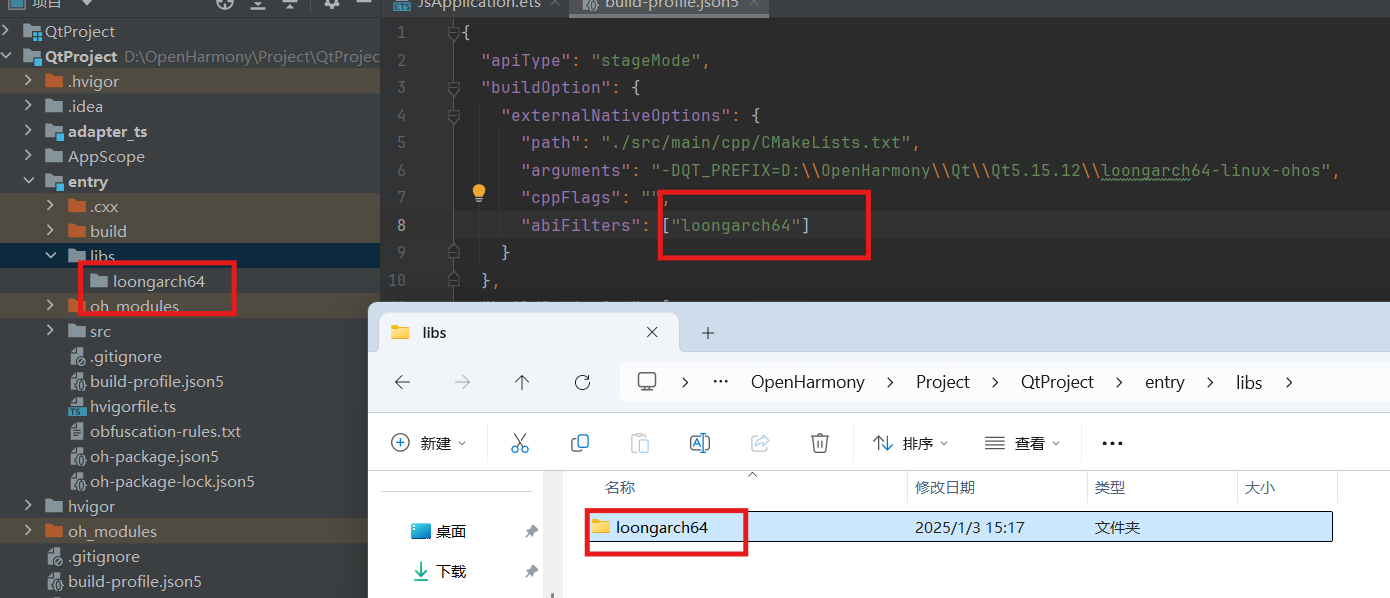


如果新建项目时module时qtmodule，这里就是libqtmodule.so。



1. 拷贝依赖：

因为Qt框架中采用了插件机制，除了拷贝依赖库到.hap包，还需要将运行时依赖的相关插件拷贝到.hap包中，在qtproject\entry\libs\目录下新建loongarch64目录，和[配置Qt开发环境](#_配置Qt开发环境)中argruments的abiFilter设置相同，如下图所示：



拷贝基础的依赖插件（libplugins\_platforms\_qopenharmony.so）到loonarch64目录中，拷贝后，再次执行[开发环境](#_开发环境_1)中【[修改项目配置](#_修改项目配置)】及【[导入产品兼容性ID](#_导入产品兼容性ID)】步骤。编译后，生成.hap包中已包含依赖的插件，如下图所示：（注：libplugins\_platforms\_qopenharmony.so插件是必要的基础插件）libplugins\_platforms\_qopenharmony.so在D:\OpenHarmony\Qt\Qt5.15.12\loongarch64-linux-ohos\plugins\platforms\目录下。

