

Building OpenHarmony

Source code, toolchains and app

Jing Zhang
Shanghai Jiao Tong University



Resources

- OpenHarmony official website: openharmony.cn
- OpenHarmony gitee: gitee.com/openharmony
- OpenHarmony download: openharmony.cn/download
- OpenHarmony application samples: openharmony/applications_app_samples
- Daily Build (binary OpenHarmony image & SDK) : [OpenHarmony CI](#)
- DevEco Studio 4.0: [OpenHarmony v4.0 Release \(2023-10-26\)](#)
- DevEco Device Tool 4.0: device.harmonyos.com/en/develop/ide

Tutorial Outline

1. Get runnable OpenHarmony binary image
2. Prepare DevEco Studio IDE and full-SDK
3. Compile HelloWorld demo application
4. Install and run HelloWorld

Environment & Marks

- Compile app in your laptop using DevEco Studio (MacOS/Windows)
- Running pre-compile OpenHarmony in our AWS server
- Running commands to manipulate OpenHarmony emulator in qemu

> stands for macOS/Windows

\$ stands for AWS Linux server

@ stands for qemu

stands for comments

Get Images

```
# download two images
$ curl -L -o oh-1.7z \
'https://ipads.se.sjtu.edu.cn:1313/f/7d8cad008a0847ba82da/?dl=1'
$ curl -L -o oh-2.7z \
'https://ipads.se.sjtu.edu.cn:1313/f/69f2c5432b9548e3b931/?dl=1'

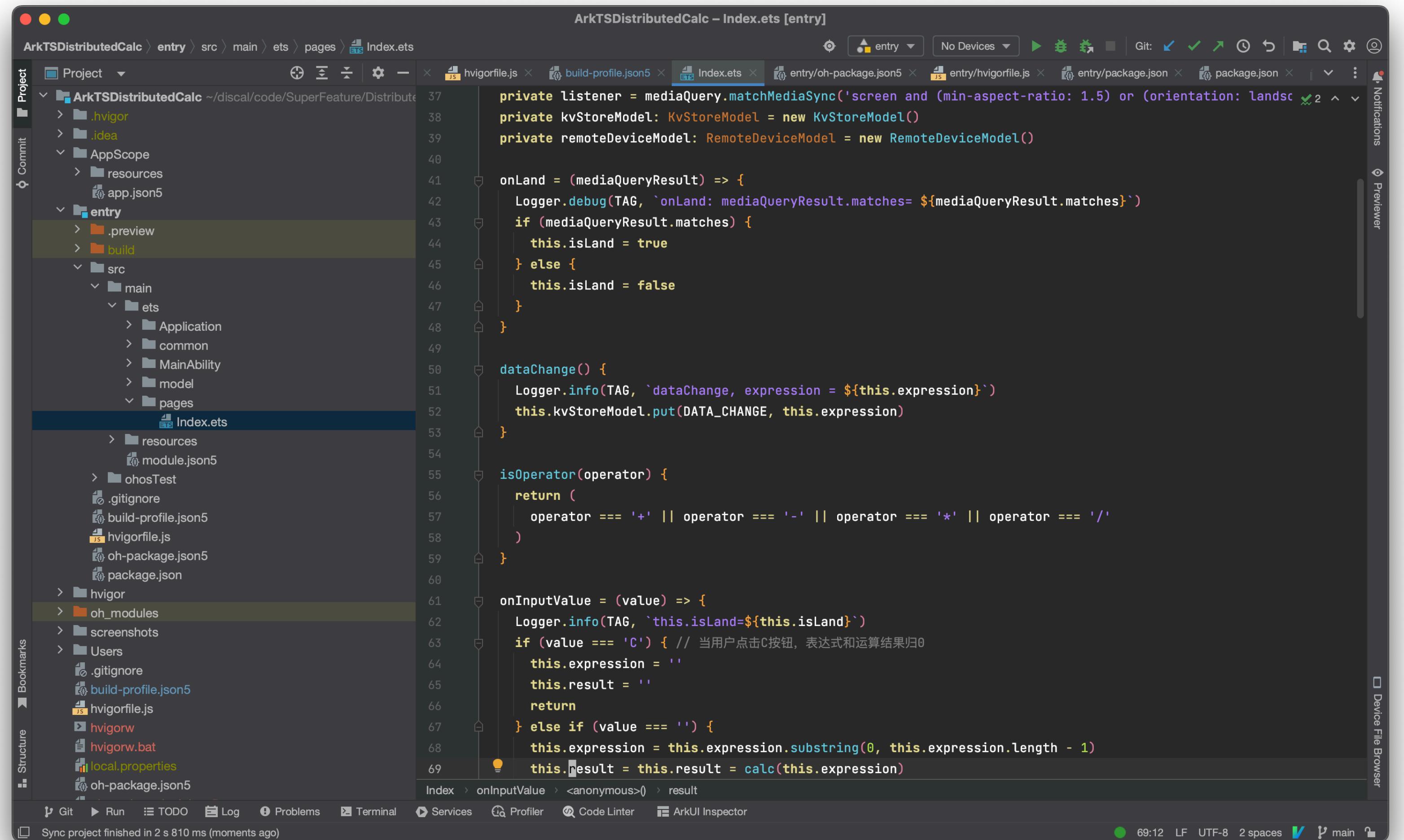
# unzip two packages
$ 7z x oh-1.7z
$ 7z x oh-2.7z
```

Download DevEco Studio IDE

Win(X64) : [Download](#)

MacOS(X64) : [Download](#)

MacOS(aarch64) :
[Download](#)



The screenshot shows the DevEco Studio IDE interface with the following details:

- Project View:** Shows the project structure under "ArkTSDistributedCalc". Key folders include ".hvigor", ".idea", "AppScope", "entry", "resources", and "src". The "src" folder contains "main", "ets", "Application", "common", "MainAbility", "model", and "pages". The "entry" folder contains "build", "resources", "module.json5", "ohosTest", ".gitignore", "build-profile.json5", "hvigorfile.js", "oh-package.json5", "package.json", "hvigor", "oh_modules", "screenshots", "Users", ".gitignore", "build-profile.json5", "hvigorfile.js", "hvigor", "hvigor.bat", "local.properties", and "oh-package.json5".
- Code Editor:** The main editor window displays the content of the "Index.ets" file, which is an ETS (Elemental UI Script) file. The code handles media queries, manages KvStoreModel and RemoteDeviceModel, and implements logic for onLand, dataChange, isOperator, and onInputValue events.
- Toolbars and Status Bar:** The top bar includes tabs for "entry", "No Devices", "Git", and other development tools. The status bar at the bottom shows "Sync project finished in 2 s 810 ms (moments ago)" and the current time "69:12".

Download Full-SDK

- Full SDK contains more system API than standard SDK
- We need to use full sdk to build distributed application

Win: [Download](#)

MacOS(aarch64): [Download](#)

MacOS(x86): [Download](#)

9
└── ets
└── js
└── native
└── previewer
└── toolchains

Replace Full-SDK in DevEco

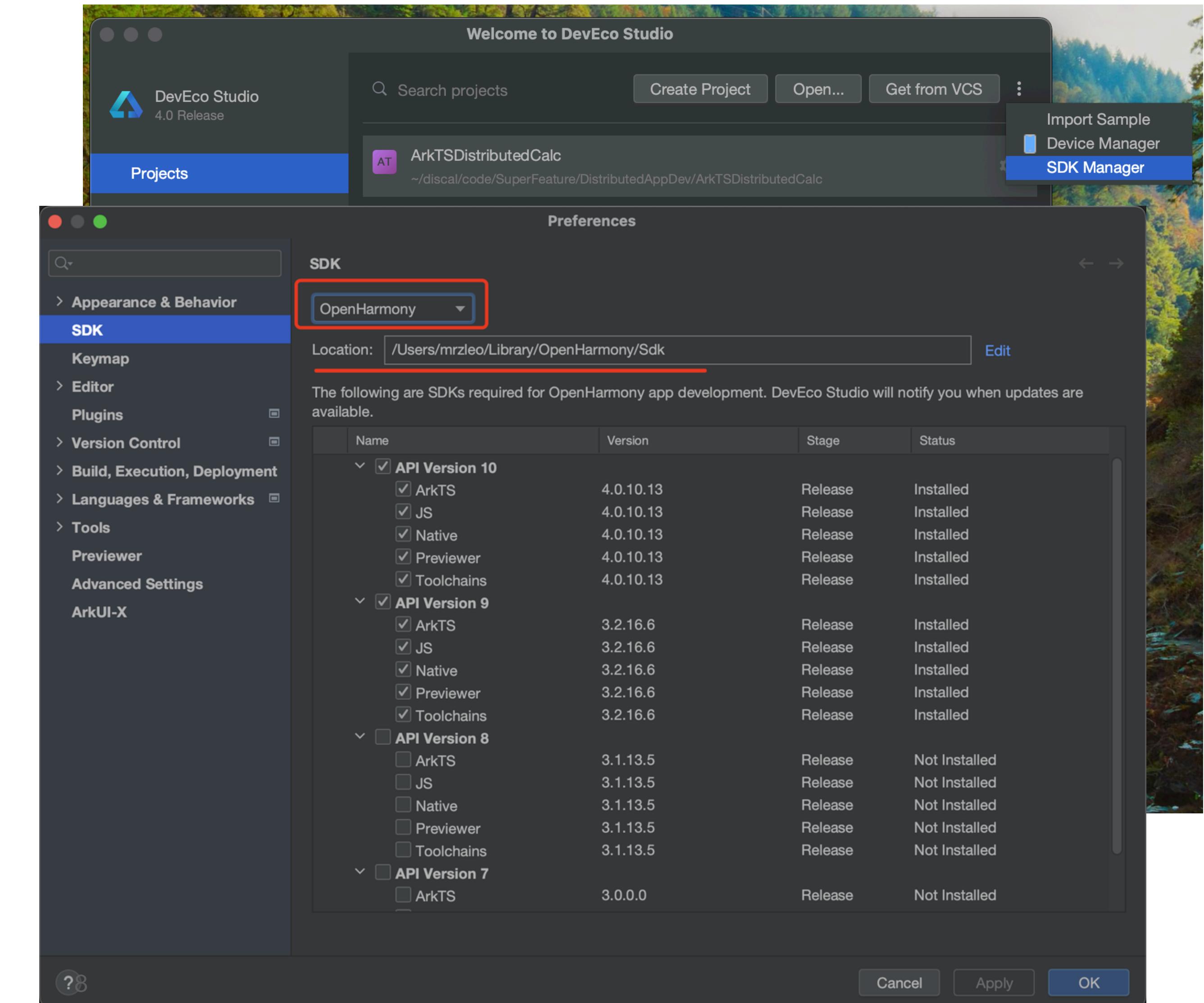
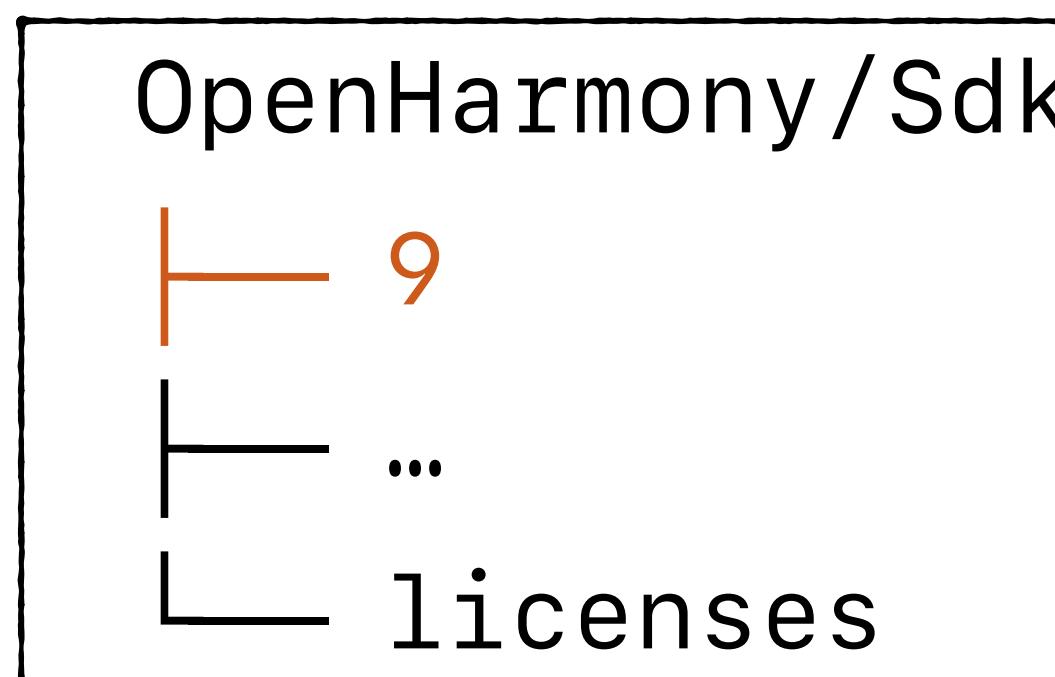
1. Click `SDK Manager` get settings

2. Find sdk path of DevEco

- “OpenHarmony”
- “Location”
- If no path, click Edit and “next”

3. Put downloaded SDK into path:

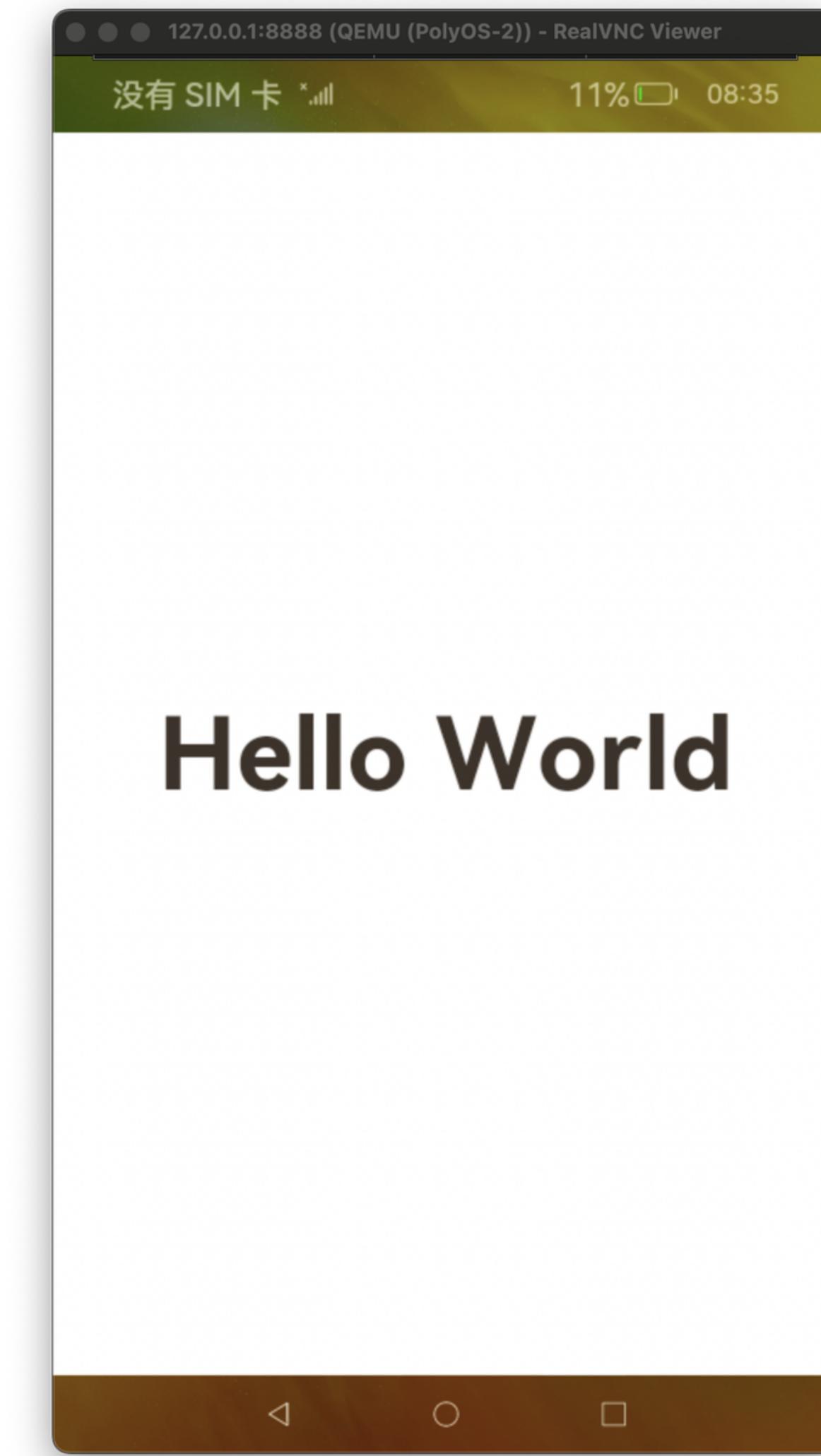
9/ -> .../OpenHarmony/Sdk/



HelloWorld Demo

- A very simple demo for OpenHarmony

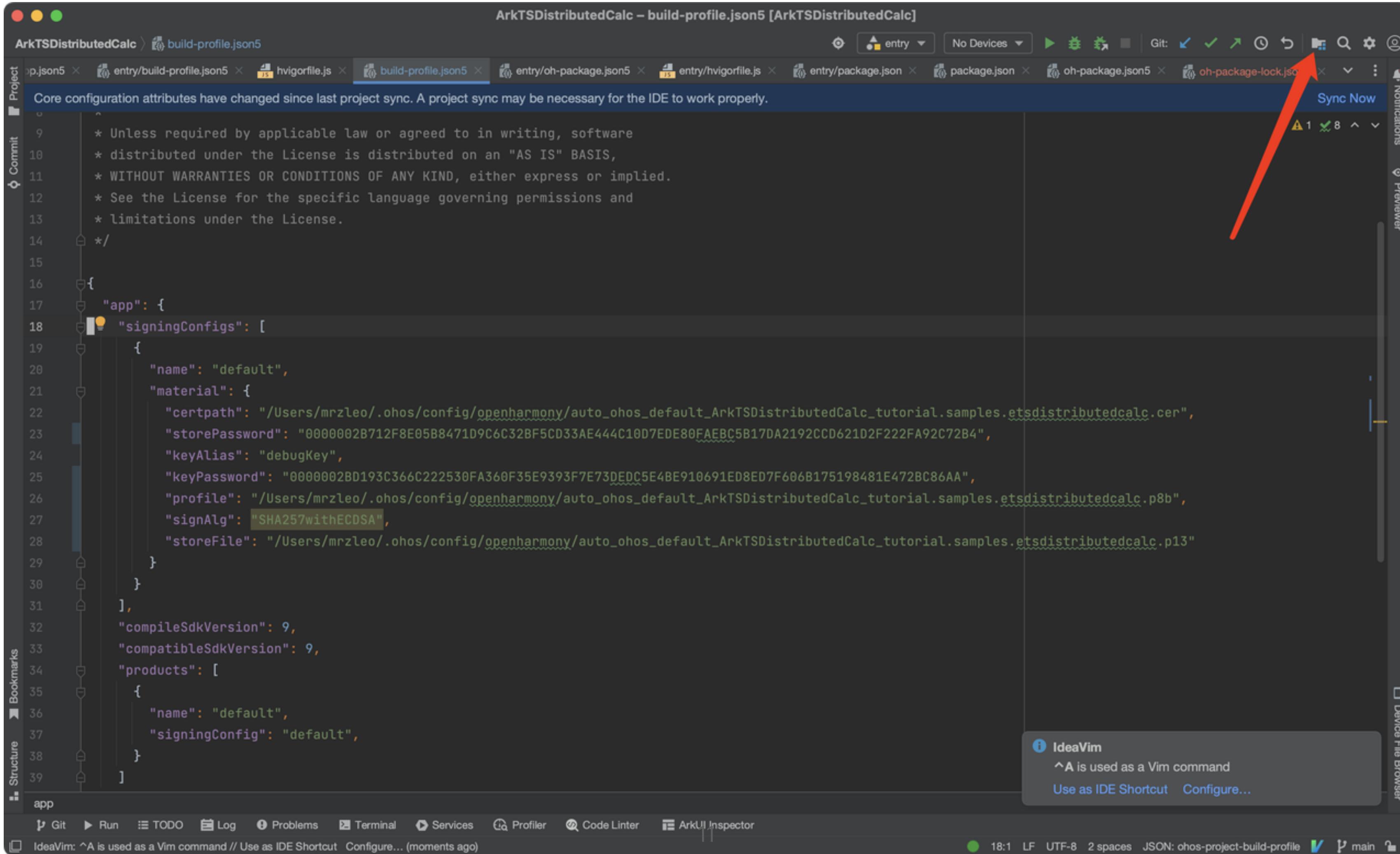
```
# download source code  
> git clone https://github.com/  
openharmony-research/helloworld-demo
```



Compile HelloWorld app

1. Generate signature
2. Compile to *Hap*
3. Install app by hdc

Generate signature



The screenshot shows an IDE interface with a dark theme. The main window displays a JSON configuration file named `build-profile.json5`. The file contains settings for a build profile, including signing configurations, compile SDK version, compatible SDK version, and products. A red arrow points to the `Sync Now` button in the top right corner of the IDE header.

```
ArkTSDistributedCalc - build-profile.json5 [ArkTSDistributedCalc]
ArkTSDistributedCalc > build-profile.json5
Project build-profile.json5 entry/build-profile.json5 h vigorfile.js build-profile.json5 entry/oh-package.json5 entry/h vigorfile.js entry/package.json package.json oh-package.json oh-package-lock.json
Core configuration attributes have changed since last project sync. A project sync may be necessary for the IDE to work properly.
{
  "app": {
    "signingConfigs": [
      {
        "name": "default",
        "material": {
          "certpath": "/Users/mrzleo/.ohos/config/openharmony/auto_ohos_default_ArkTSDistributedCalcTutorial.samples.etsdistributedcalc.cer",
          "storePassword": "0000002B712F8E05B8471D9C6C32BF5CD33AE444C10D7EDE80FAEBC5B17DA2192CCD621D2F222FA92C72B4",
          "keyAlias": "debugKey",
          "keyPassword": "0000002BD193C366C222530FA360F35E9393F7E73DEDCE5E4BE910691ED8ED7F606B175198481E472BC86AA",
          "profile": "/Users/mrzleo/.ohos/config/openharmony/auto_ohos_default_ArkTSDistributedCalcTutorial.samples.etsdistributedcalc.p8b",
          "signAlg": "SHA257withECDSA",
          "storeFile": "/Users/mrzleo/.ohos/config/openharmony/auto_ohos_default_ArkTSDistributedCalcTutorial.samples.etsdistributedcalc.p13"
        }
      }
    ],
    "compileSdkVersion": 9,
    "compatibleSdkVersion": 9,
    "products": [
      {
        "name": "default",
        "signingConfig": "default"
      }
    ]
  }
}
```

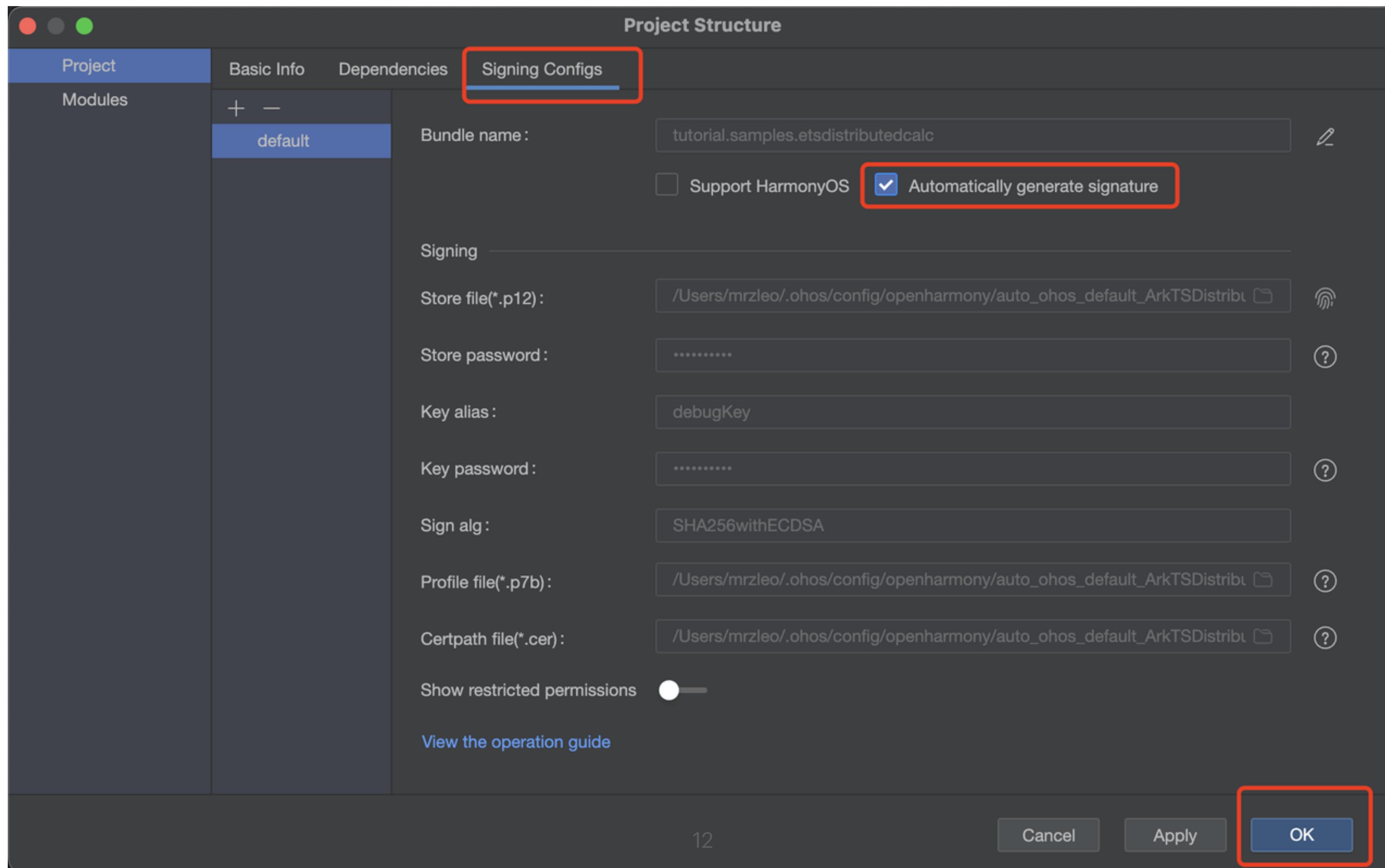
IdeaVim
^A is used as a Vim command
Use as IDE Shortcut Configure...

Git Run TODO Log Problems Terminal Services Profiler Code Linter ArkUI Inspector

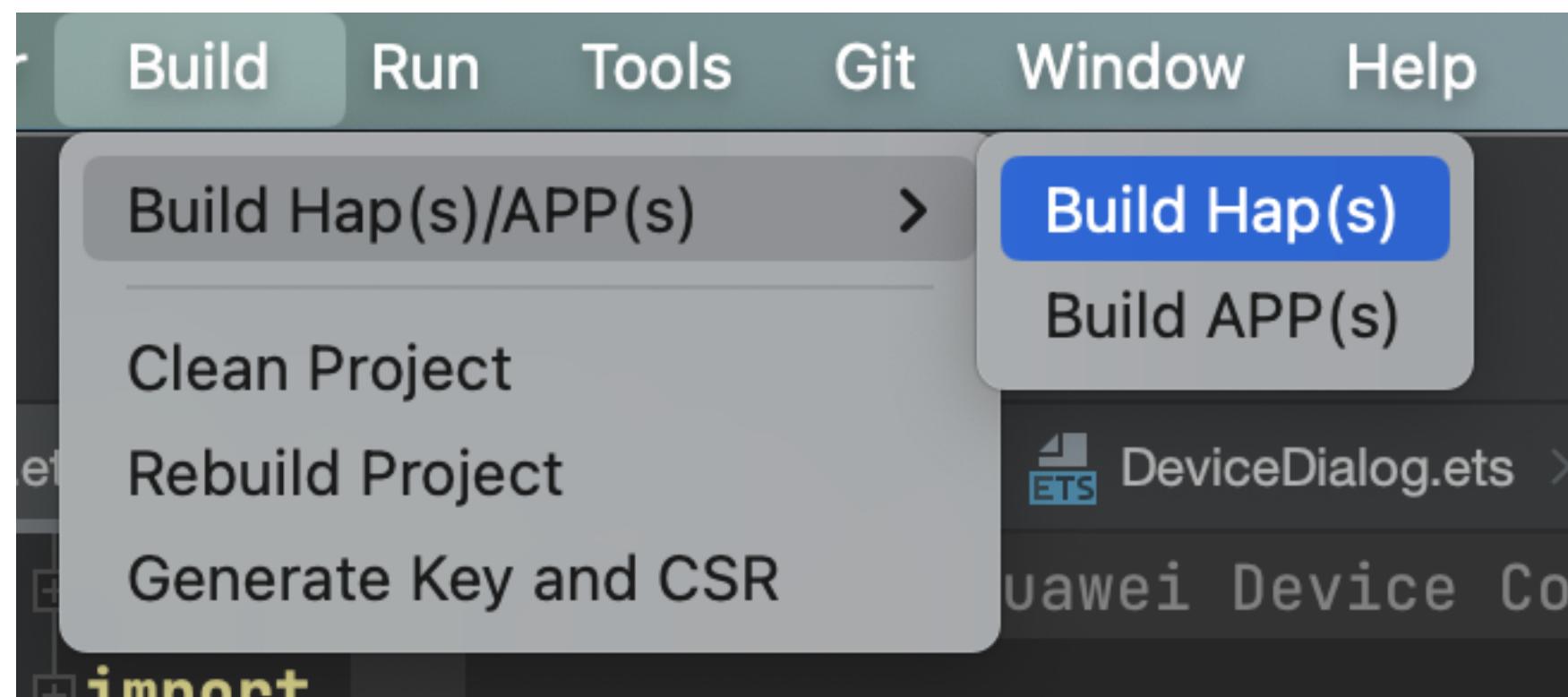
IdeaVim: ^A is used as a Vim command // Use as IDE Shortcut Configure... (moments ago)

18:1 LF UTF-8 2 spaces JSON: ohos-project-build-profile ✓ main

Generate signature



Compile to Hap



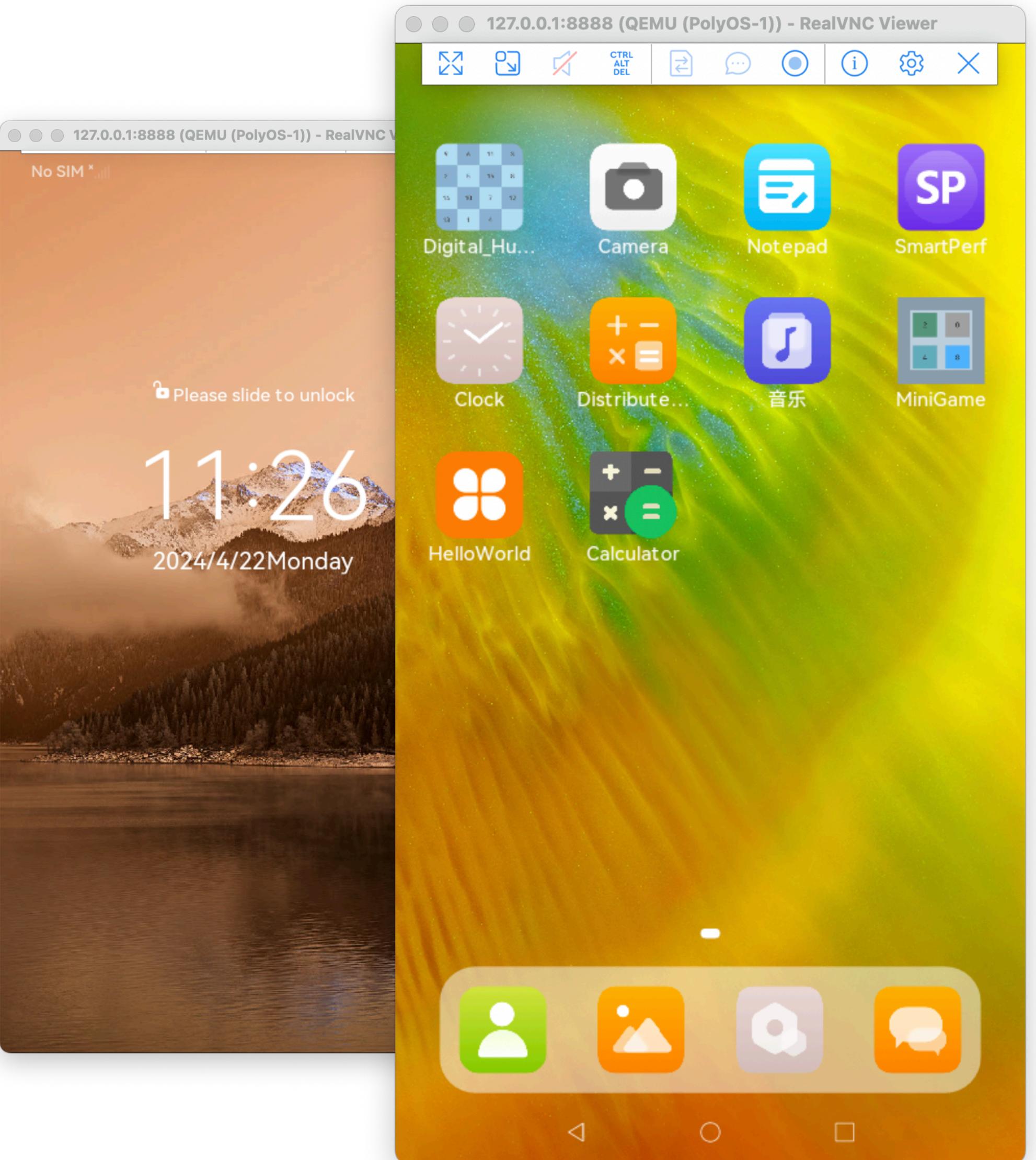
```
> hvigor UP-TO-DATE :entry:default@MergeProfile...
> hvigor UP-TO-DATE :entry:default@GenerateLoaderJson...
> hvigor Finished :entry:default@BuildNativeWithCmake... after 1 ms
> hvigor UP-TO-DATE :entry:default@MakePackInfo...
> hvigor UP-TO-DATE :entry:default@ProcessProfile...
> hvigor Finished :entry:default@BuildNativeWithNinja... after 1 ms
> hvigor UP-TO-DATE :entry:default@ProcessResource...
> hvigor UP-TO-DATE :entry:default@ProcessLibs...
> hvigor UP-TO-DATE :entry:default@CompileResource...
> hvigor UP-TO-DATE :entry:default@CompileArkTS...
> hvigor Finished :entry:default@BuildJS... after 6 ms
> hvigor UP-TO-DATE :entry:default@PackageHap...
> hvigor UP-TO-DATE :entry:default@SignHap...
> hvigor Finished :entry:assembleHap... after 1 ms
> hvigor BUILD SUCCESSFUL in 230 ms

Process finished with exit code 0
```

```
# Check output
> ls entry/build/default/outputs/default/entry-default-signed.hap
entry/build/default/outputs/default/entry-default-signed.hap
```

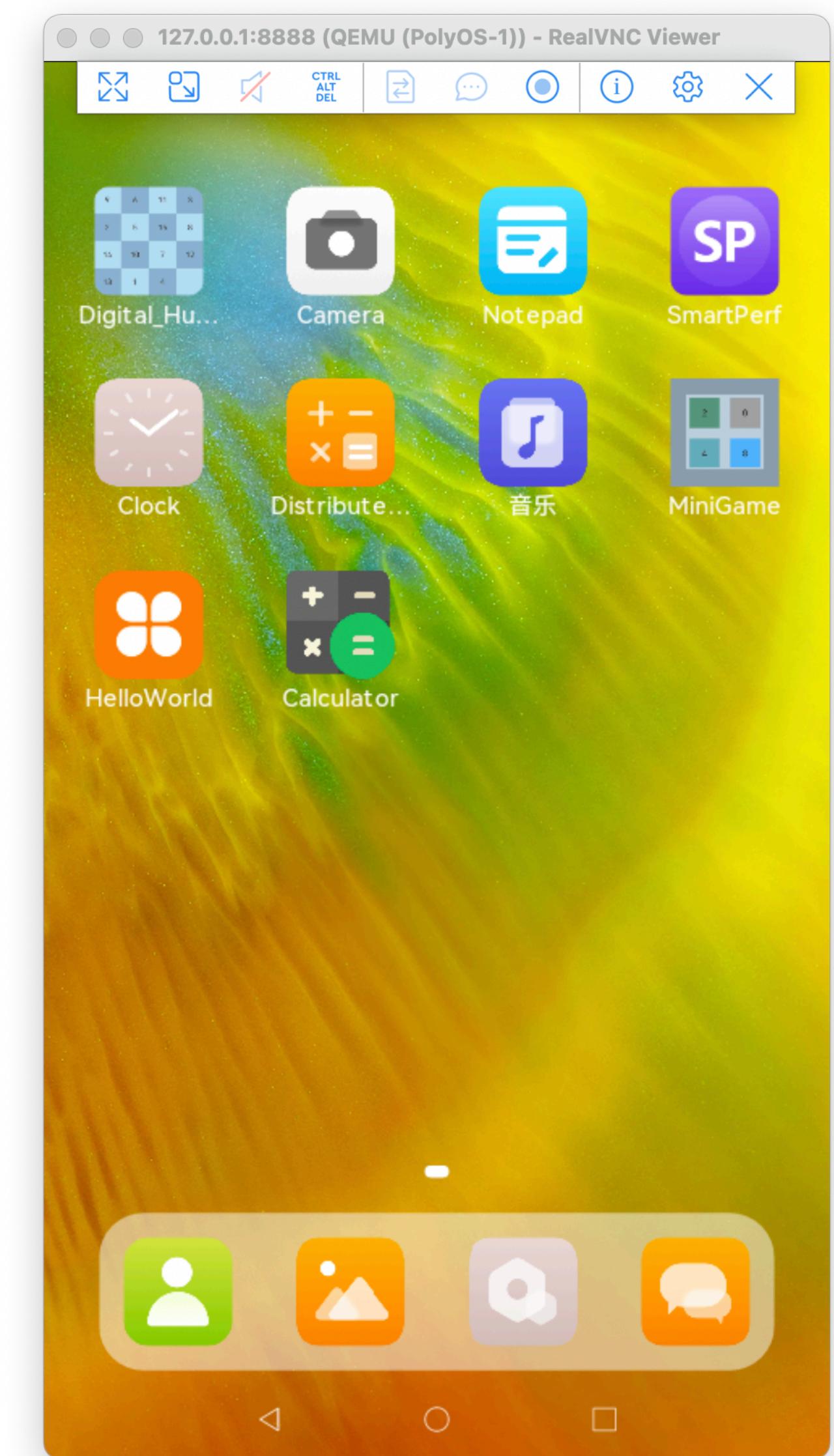
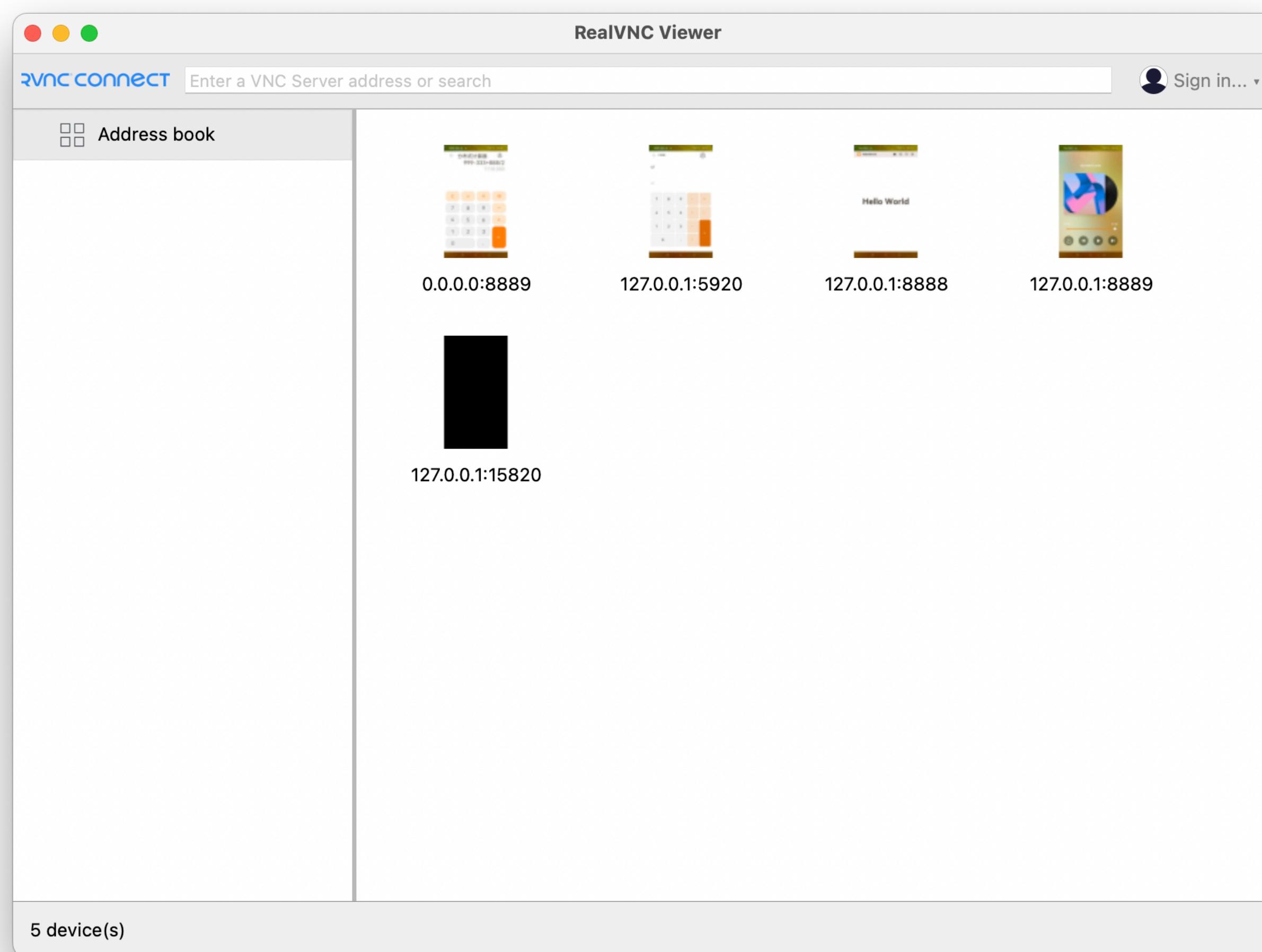
Install App

```
# send to server  
> scp entry/build/default/outputs/default/  
entry-default-signed.hap admin@asplos-1:/  
home/admin/helloworld.hap  
  
# boot OpenHarmony emulator, forward ip  
port to localhost using ssh  
> ssh admin@asplos-1 -L 8888:localhost:5920  
$ cd ~/oh-1  
$ ./boot.sh
```



Connect to VNC server

```
# use any VNC client to connect to the  
emulator(localhost:8888) for screen
```



Install in OpenHarmony

```
# connect to emulator using hdc  
$ hdc tconn 192.168.137.3:5555
```

```
# Install app using hdc  
$ hdc -t 192.168.137.3:5555 \  
install ~/helloworld.hap
```



Run HelloWorld demo

```
# connect to openharmony shell  
$ hdc -t 192.168.137.3:5555 shell  
  
# emulate user input, unlock the screen  
@ uinput -T -d 0 0 -m 0 0 0 800 -u 0 800  
  
# start HelloWorld demo  
@ aa start -a EntryAbility \  
    -b com.example.helloworld  
  
# exit helloworld demo  
@ aa force-stop com.example.helloworld
```

