

A Tool for Converting MiniApp Manifest to Harmony OS JS FA Config file

Zitao Wang (wangzitao@huawei.com)

Shouren Lan (lanshouren@huawei.com)

Zhouze Li (lizhouze@huawei.com)

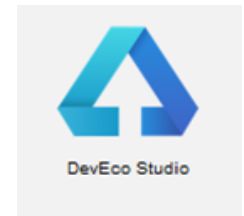
Our Goal

- Develop a Tool for Converting MiniApp Manifest to Harmony OS JS FA Config file,
 - to verify the feasibility of the standard
 - Enables cross-platform deployment of standard MiniApp
 - Improve developers' confidence in standard development methods.



- Development Environment

- DevEco Studio 3.0 Beta1 (https://developer.harmonyos.com/cn/develop/deveco-studio#download_beta)
 - A one-stop, distributed platform that enables you to develop versatile all-device, all-scenario apps, offering distributed multi-device development, debugging, and emulation, as well as comprehensive quality and security safeguards.
- Node.js
 - an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.



What is Harmony OS JS FA? What is Config file?

- HarmonyOS JS FA

- HarmonyOS JS FA is a kind of MiniApp, that relies on HarmonyOS instead of super app.
- Its framework supports distributed hopping.
 - Service hopping is a distributed operation mode of HarmonyOS.
 - The hop capability breaks boundaries of devices and enables multi-device collaboration, universal coordination, and seamless hops of atomic services. For example, a user can edit the same email, play a game across devices, etc.



A user uses a mobile phone as a controller to play games on HUAWEI Vision.

- Config file

- The root directory for each HAP file of an App contains a config.json file. It's similar to Manifest and provides the following information:
 - Describe the app, such as the bundle name, version number, and vendor
 - Device-specific configuration information, including the backup and restoration as well as network security capabilities of the application.
 - Configuration information of the HAP file, including the basic attributes that must be defined for each ability (such as the bundle name, class name, and type) as well as the permissions required by the application to access a protected part of the system or another application

```
C:\Users\w00289200\DevEcoStudioProjects\MyAppli 1
> .gradle 2
> .idea 3
> entry 4
  libs 5
  src 6
    main 7
      java 8
      js 9
      resources 10
      config.json 11
    ohosTest 12
  .gitignore 13
  build.gradle 14
  package.json 15
  proguard-rules.pro
```

```
1  "app": {
2  "bundleName": "com.example.myapplication",
3  "vendor": "example",
4  "version": {
5  "code": 1000000,
6  "name": "1.0.0"
7  },
8  },
9  "deviceConfig": {},
10 "module": {
11 "package": "com.example.myapplication",
12 "name": ".MyApplication",
13 "mainAbility": "com.example.myapplication.MainAbility",
14 "deviceType": [
15
```

The Gaps and design methods

Same attribute, same name

Standard MiniApp Manifest	HOS JS FA Config.json
Version	Version
Icons	Icons
Pages	Pages
Window	window

Same Meaning, Different Naming

Standard MiniApp Manifest	HOS JS FA Config.json
Dir	Orientation
App_id	Bundle_name
Platform_version	apiVersion
Req_permissions	reqPermissions
Widgets	form

Unsupported attribute

Standard MiniApp Manifest	HOS JS FA Config.json
Lang	Implicit support
Name	Implicit support
Description	

Design Methods

1. For the first set, directly fulfill the value.
2. For the second set, extract the attribute value from the standard manifest file and fill the corresponding attribute in the config.json.
3. For the third set,
 1. extract the attribute value from the standard manifest and fill it in the field that implicitly supports this function in the config.json.
 2. discard the attributes.

Demo

- Input a standard MiniApp Manifest file, and create a new HarmonyOS project.
- Running the conversion script
- The converted Results.
- Running the converted App.

Standard MiniApp Manifest

```
{
  "dir": "ltr",
  "lang": "en-US",
  "app_id": "org.example.miniapp",
  "name": "Miniapp",
  "short_name": "Miniapp",
  "version": {
    "name": "1.0.1",
    "code": 11
  },
  "description": "A Simple MiniApp Demo",
  "icons": [
    {
      "src": "common/icon.png",
      "sizes": "48*48"
    }
  ],
  "platform_version": {
    "min_code": 6,
    "target_code": 6,
    "release_type": "Beta1"
  },
  "pages": [
    "pages/index/index"
  ],
  "window": {
    "navigation_bar_text_style": "black",
    "navigation_bar_title_text": "My MiniApp",
    "navigation_bar_background_color": "#f8f8f8",
    "background_color": "#ffffff",
    "fullscreen": false
  },
  "req_permissions": [
    {
      "name": "system.permission.LOCATION",
      "reason": "To show user's position on the map"
    }
  ]
}
```

HOS JS FA Config.json

```
{
  "app": {
    "bundleName": "com.example.myapplication",
    "vendor": "example",
    "version": {
      "code": 1000000,
      "name": "1.0.0"
    }
  },
  "deviceConfig": {},
  "module": {
    "package": "com.example.myapplication",
    "name": ".MyApplication",
    "mainAbility": "com.example.myapplication.MainAbility",
    "deviceType": [
      "phone",
      "tablet",
      "tv",
      "wearable"
    ],
    "distro": {
      "deliveryWithInstall": true,
      "moduleName": "entry",
      "moduleType": "entry",
      "installationFree": true
    },
    "abilities": [
      {
        "skills": [
          {
            "entities": [
              "entity.system.home"
            ],
            "actions": [
              "action.system.home"
            ]
          }
        ],
        "visible": true,
        "name": "com.example.myapplication.MainAbility"
      }
    ]
  }
}
```

convert



Converted HOS JS FA Config.json

```
{
  "app": {
    "bundleName": "org.example.miniapp",
    "vendor": "example",
    "version": {
      "name": "1.0.1",
      "code": 11
    },
    "apiVersion": {
      "compatible": 6,
      "target": 6,
      "releaseType": "Beta1"
    }
  },
  "deviceConfig": {},
  "module": {
    "package": "com.example.myapplication",
    "name": ".MyApplication",
    "mainAbility": "com.example.myapplication.MainAbility",
    "deviceType": [
      "phone",
      "tablet",
      "tv",
      "wearable"
    ],
    "distro": {
      "deliveryWithInstall": true,
      "moduleName": "entry",
      "moduleType": "entry",
      "installationFree": true
    },
    "abilities": [
      {
        "skills": [
          {
            "entities": [
              "entity.system.home"
            ],
            "actions": [
              "action.system.home"
            ]
          }
        ],
        "visible": true,
        "name": "com.example.myapplication.MainAbility"
      }
    ]
  }
}
```

Running Converted App



The Next Steps

- Continue to develop converting tools to support complete applet standards, including lifecycles, package, UI component, etc.
- Open Code:
 - Establish a project and contribute code in the open source community, such as OpenHarmony.
 - Liaise with relevant industry organizations, such as Software Green Alliance.
 - Integrate the converting tools into the HOS IDE.

The Crews

Ordered by Last Name

- Shouren Lan
- Zhouze Li
- Zitao Wang (Speaker)
- Zhiqiang Yu
- Yongjing Zhang