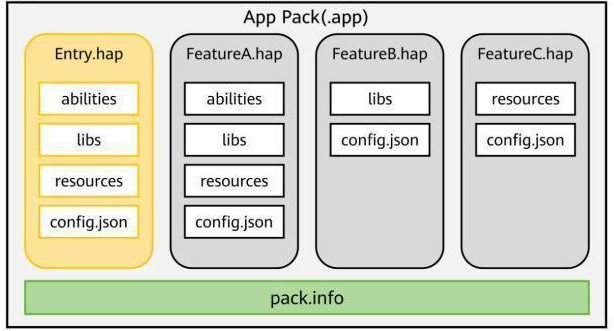
* HarmonyOS用户应用程序包以APP Pack（Application Package）形式发布，它是由一个或多个HAP（HarmonyOS Ability Package）以及描述每个HAP属性的pack.info组成的
* HAP是Ability的部署包，HarmonyOS应用代码围绕Ability组件展开
* HAR：HarmonyOS Ability Resources
  + HAR可以提供构建应用所需的所有内容，包括源代码、资源文件和config.json文件
  + HAR不同于HAP，HAR不能独立安装运行在设备上，只能作为应用模块的依赖项被引用



* HAP由以下四个部分组成：
  + 代码（abilities）
  + 第三方库（libs）
  + 资源（resources）
  + 应用配置文件（config.json）
* HAP有两个种类，分别是：
  + entry：应用的主模块
    - 一个APP中，对于同一设备类型必须有且只有一个entry类型的HAP，可独立安装运行
    - 在src/main/java下以包名命名的文件夹内分布着Java代码。这里的代码可以用来创建布局，动态调整布局以及为交互提供支撑服务
    - 和java文件夹同级的resources目录下分布着应用资源，该目录下的base目录下，按资源用途又分为多个文件夹资源：
      * element：表示元素资源。该文件夹下主要存放json格式的文件，主要用来表示字符串、颜色值、布尔值等，可以在其他地方被引用
      * graphic：表示可绘制资源，用xml文件来表示。比如我们项目中设置的圆角按钮、按钮颜色等都是通过引用这里的资源来统一管理的
      * layout：表示布局资源，用xml文件来表示。比如页面的布局资源，都放在这里
      * media：表示媒体资源。包括图片、音频、视频等非文本格式的文件
  + feature：应用的动态特性模块
    - 一个APP可以包含一个或多个feature类型的HAP，也可以不含
* 只有包含Ability的HAP才能够独立运行
* 库文件
  + 库文件是应用依赖的第三方代码（例如so，jar，bin，har等二进制文件），存放在libs目录
* 资源文件
  + 应用的资源文件（字符串、图片、音频等）存放在resources目录下
  + resources目录包括两大类目录
    - base目录（默认存在的目录）与限定词目录
      * 限定词目录（需要开发者自行创建）：由一个或多个表征应用场景或设备特征的限定词组合而成，包括语言、文字、国家或地区、横竖屏、设备类型和屏幕密度等六个维度，限定词之间通过下划线或中划线连接
    - rawfile目录（默认存在的目录）
  + 资源组目录：base 目录与限定词目录下面可以创建资源组目录（包括 element、media、animation、layout、graphic、profile），用于存放特定类型的资源文件
  + 使用资源文件
    - base目录与限定词目录中的资源文件：通过指定资源类型（type）和资源名称（name）来引用
      * Java文件引用资源文件的格式：ResourceTable.type\_name
        + 特别地，如果引用的是系统资源，则采用：ohos.global.systemres.ResourceTable.type\_name
      * XML文件引用资源文件的格式：$type:name
        + 如果引用的是系统资源，则采用：$ohos:type:name
      * rawfile目录中的资源文件：通过指定文件路径和文件名称来引用

# 配置文件

* 配置文件 (config.json) 是应用的Ability信息，用于声明应用的Ability，以及应用所需权限等信息
* 该配置文件位于：entry/src/main目录下，由工具帮我们生成，命名为config.json
* 在配置文件 (config.json) 中注册 Ability 时，可以通过配置 Ability元素中的 type 属性来指定 Ability 模板类型，其中，type 的取值可以为 page、service 或 data
* 应用的每个HAP的根目录下都存在一个“config.json”配置文件，文件内容主要涵盖以下三个方面（config.json配置文件中的一级目录）：
  + app
  + deviceconfig
  + module
* 配置文件的内部结构：“config.json”由“app”、“deviceConfig”和“module”三个部分组成，缺一不可

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 含义 | 数据类型 | 是否可缺省 |
| app | * 表示应用的全局配置信息 * 包含应用的包名、生产厂商、版本号等基本信息 * 同一个应用的不同HAP包的“app”配置必须保持一致 | 对象 | 否 |
| deviceconfig | * 表示应用在具体设备上的配置信息 * 包含应用的备份恢复、网络安全等能力 |
| module | * 表示HAP包的配置信息 * 包含每个Ability必须定义的基本属性（如包名、类名、类型以及Ability提供的能力），以及应用访问系统或其他应用受保护部分所需的权限等 * 该标签下的配置只对当前HAP包生效 |

### app

* bundleName：表示应用的包名，用于标识应用的唯一性。通常采用反转的域名（示（例如，com.huawei.himusic）。建议第一级为域名后缀com，第二级为厂商/个人名，第三级为应用名，也可以采用多级。）
* vendor：表示对开发应用的厂商的描述
* version（表示应用的版本信息）：
  + code表示内部版本号，用于系统管理版本使用，对用户不可见
  + name表示应用的版本号，用于向用户呈现
* apiVersion（表示应用依赖的HarmonyOS的API版本）：
  + compatible：表示应用运行需要的API最小版本，取值为大于零的整数
  + target：
    - 表示应用运行需要的API目标版本，取值为大于零的整数
    - 可缺省，缺省值为应用所在设备的当前API版本
  + releaseType：表示应用运行需要的API目标版本的类型，取值为以下三种（N代表大于零的整数）（可缺省，缺省值为Release）
    - CanaryN：受限发布的版本
    - BetaN：公开发布的Beta版本
    - ReleaseN：公开发布的正式版本

### deviceconfig

* deviceConfig 包含在具体设备上的应用配置信息，包含以下属性：
  + default：
    - 表示所有设备通用的应用配置信息
    - 不可缺省
    - default 标签内的配置是适用于所有设备通用，其他设备类型如果有特殊的需求，则需要在该设备类型的标签下进行配置
  + phone：可缺省，缺省为空
  + tablet：
    - 表示平板的应用配置信息
    - 可缺省，缺省为空
  + tv：可缺省，缺省为空
  + car：可缺省，缺省为空
  + wearable：可缺省，缺省为空
  + liteWearable：可缺省，缺省为空
  + smartVision：
    - 表示智能摄像头特有的应用配置信息
    - 可缺省，缺省为空

### module

* supportedModes：表示应用支持的运行模式。当前只定义了驾驶模式（drive）。该标签仅适用于车机
* deviceType：
  + 表示允许Ability运行的设备类型
  + 系统预定义的设备类型包括：
    - phone（手机）
    - tablet（平板）
    - tv（智慧屏）
    - car（车机）
    - wearable（智能穿戴）
    - liteWearable（轻量级智能穿戴）
    - …………
* …………

### 配置文件的配置项组成

* 配置文件“config.json”采用JSON文件格式，其中包含了一系列配置项，每个配置项由属性和值两部分构成
  + 属性：属性出现顺序不分先后，且每个属性最多只允许出现一次
  + 值：每个属性的值为JSON的基本数据类型（数值、字符串、布尔值、数组、对象或者null类型）
* DevEco Studio 提供了两种编辑 config.json 文件的方式。在 config.json 的编辑窗口中，可在右上角切换代码编辑视图或可视化编辑视图

# pack.info

* 描述应用软件包中每个HAP的属性，由IDE编译生成，应用市场根据该文件进行拆包和HAP的分类存储
* HAP的具体属性包括
  + delivery-with-install：
    - 表示该HAP是否支持随应用安装
    - “True”表示支持随应用安装
    - “False”表示不支持
  + name：HAP文件名
  + module-type：
    - 模块类型entry或feature
  + device-type：表示支持该HAP运行的设备类型