



本章介绍HarmonyOS产生的历史背景、核心技术理念及开发环境的搭建，同时介绍了HarmonyOS NEXT的新特性，并演示了如何通过DevEco Studio来初始化HarmonyOS项目结构和创建一个简单的应用程序。

1.1 HarmonyOS 产生的背景

2024年10月8日，HarmonyOS NEXT Release版本正式发布，标志着以API 12为核心的HarmonyOS全套开发套件（含SDK及开发工具DevEco Studio）均达到Release状态并正式发布。开发者可基于Release状态的开发套件进行应用开发并正式上架华为应用市场。

同日，HarmonyOS NEXT开启公测，首批公测机型包括华为Mate 60系列、华为Mate X5系列、华为MatePad Pro 13.2英寸系列。

那么到底什么是HarmonyOS？为什么需要HarmonyOS？

1.1.1 万物互联时代的新挑战

经过十多年的发展，传统移动互联网的增长红利已渐见顶。万物互联时代正在开启，应用的设备底座将从几十亿手机扩展到数百亿物联网（Internet of Things, IoT）设备。全新的全场景设备体验，正深入改变消费者的使用习惯。同时应用开发者也面临设备底座从手机单设备到全场景多设备的转变，通过全场景多设备作为全新的底座，为消费者带来万物互联时代更为高效、便捷的体验。

新的场景同时也带来了新的挑战。开发者不仅需要支持更加多样化的设备，还需要支持跨设备的协作。不同设备类型意味着不同的传感器能力、硬件能力、屏幕尺寸、操作系统和开发语言，还意味着差异化的交互方式。同时跨设备协作也让开发者面临分布式开发带来的各种复杂性，例如跨设备的网络通信、数据同步等。若采取传统开发模式，适配和管理工作量将非常巨大。当前移动应用开发中遇到的主要挑战包括：

- 针对不同设备上的不同操作系统，重复开发，维护多套版本。
- 多种语言栈，对开发人员技能要求高。
- 多种开发框架，不同的编程范式。
- 命令式编程，需关注细节，变更频繁，维护成本高。

与此同时，AI时代全面来临，在PC互联网到移动互联网再到智能化终端的演进过程中，AI计算

主要在云端数据中心进行，非常依赖网络，具有一定的延时，且数据传输的安全性、私密性不能得到有效保证。随着人们对交互和信息获取的智能化要求越来越高，移动设备的计算能力越来越强，在设备侧就能提供AI的相关能力，例如自然语言交互、环境智能感知、图像识别等。如何快速地使用设备侧的强大AI能力，使自己的应用更加智能化，进而更好地服务消费者，也是开发者面临的全新挑战。

为了更好地抓住机遇，应对万物互联所带来的一系列挑战，新的应用生态应该具备如下特征。

- 单一设备延伸到多设备：应用一次开发就能在多个设备上运行，软件实体能够从单一设备转移到其他设备上，且多个设备间能够协同运行，给消费者提供全新的分布式体验。
- 厚重应用模式到轻量化服务模式：提供轻量化的服务较低的资源消耗，一步直达，快速完成消费者特定场景的任务。
- 集中化分发到AI加持下的智慧分发：为消费者提供智慧场景服务，实现“服务找人”。
- 纯软件到软硬芯协同的AI能力：提供软硬芯协同优化的原生AI能力，全面满足应用高性能诉求。

因此，在这个背景下，华为推出了自己的操作系统——HarmonyOS。正如其中文“鸿蒙”的寓意，意味着这个HarmonyOS将会开启一个开天辟地的时代，2020年12月16日，华为发布HarmonyOS 2.0手机开发者Beta版本，这意味着HarmonyOS能够覆盖手机应用场景。2022年11月4日，HarmonyOS 3.1开发者尝鲜版本发布，以支撑声明式开发体系。2023年5月16日，HarmonyOS 3.1 Release版本正式发布。2023年8月4日，HarmonyOS 4和HarmonyOS NEXT开发者预览版本正式发布。2024年1月18日，HarmonyOS NEXT开发者预览版（鸿蒙星河版）面向开发者开放申请。2024年6月21日，正式面向开发者启动HarmonyOS NEXT Beta版本。2024年10月8日，HarmonyOS NEXT Release版本正式发布。

1.1.2 什么是 HarmonyOS

HarmonyOS在2019年8月9日华为为开发者大会上首次公开亮相，时任华为消费者业务CEO余承东进行了关于HarmonyOS的主题演讲。

HarmonyOS也称为鸿蒙系统，或者鸿蒙OS，这是一款面向万物互联时代的、全新的分布式操作系统。在传统的单设备系统能力基础上，HarmonyOS提出了基于同一套系统能力、适配多种终端形态的分布式理念，能够支持手机、平板电脑、智能穿戴、智慧屏、车机、PC、智能音箱、耳机、AR/VR眼镜等多种终端设备，提供全场景（移动办公、运动健康、社交通信、媒体娱乐等）业务能力。

- 对消费者而言，HarmonyOS用一个统一的软件系统，从根本上解决了消费者使用大量终端体验割裂的问题。HarmonyOS能够将生活场景中的各类终端进行能力整合，可以实现不同的终端设备之间的快速连接、能力互助、资源共享，匹配合适的设备，为消费者提供统一、便利、安全、智慧化的全场景体验。
- 对应用开发者而言，HarmonyOS采用了多种分布式技术，整合各种终端硬件能力，形成一个虚拟的“超级终端”。开发者可以基于“超级终端”进行应用开发，使得应用程序的开发实现与不同终端设备的形态差异无关。这能够让开发者聚焦上层业务逻辑，无须关注硬件差异，更加便捷、高效地开发应用。
- 对设备开发者而言，HarmonyOS采用了组件化的设计方案，可以按需调用“超级终端”能力，并带来“超级终端”的创新体验。根据设备的资源能力和业务特征进行灵活剪裁，满足不同形态的终端设备对于操作系统的需求。

举例来说，当用户走进厨房，用HarmonyOS手机触碰微波炉，就能实现设备极速联网；用HarmonyOS手机触碰豆浆机，就能快速实现无屏变有屏。

自HarmonyOS诞生以来，经过多年的发展，终于迎来了HarmonyOS NEXT版本。HarmonyOS NEXT也带来了更多惊喜，全新推出应用开发Stage模型，并在ArkTS语言、应用程序框架、Web、ArkUI等子系统能力方面有所更新或增强。

1.1.3 HarmonyOS 应用开发

为了进一步扩大HarmonyOS的生态圈，面对广大的硬件设备厂商，HarmonyOS通过SDK、源代码、开发板/模组和HUAWEI DevEco Studio等装备共同构成了完备的开发平台与工具链，让HarmonyOS设备开发易如反掌。

应用创新是一款操作系统发展的关键，应用开发体验更是如此。一条完整的应用开发生态中，应用框架、编译器、IDE、API/SDK都是必不可少的。为了赋能开发者，HarmonyOS提供了一系列构建全场景应用的完整平台工具链与生态体系，助力开发者，让应用能力可分可合可流转，轻松构筑全场景创新体验。

可以预见的是，HarmonyOS必将是近些年的热门话题。对于能在早期投身于HarmonyOS开发的技术人员而言，其意义不亚于当年早期Android的开发。HarmonyOS必将带给开发者广阔的前景。同时，基于HarmonyOS所提供的完善的平台工具链与生态体系，相信广大的读者一定也能轻松入门HarmonyOS。

5G网络准备就绪，物联网产业链也已经渐趋成熟，在物联网即将爆发的前夜，正急需一套转为物联网准备的操作系统，华为的HarmonyOS正逢其时。Windows系统成就了微软，Android系统成就了谷歌，HarmonyOS系统是否能成就华为，让我们拭目以待。

1.2 HarmonyOS 核心技术理念

在万物智联时代重要机遇期，HarmonyOS结合移动生态发展的趋势，提出了三大技术理念(见图1-1)：一次开发多端部署、可分可合自由流转、统一生态原生智能。



图 1-1 核心技术理念

1.2.1 一次开发，多端部署

“一次开发，多端部署”指的是一个工程，一次开发上架，多端按需部署。其目的是为了支撑开发者高效地开发多种终端设备上的应用。为了实现这一目的，HarmonyOS提供了几个核心能力，包括多端开发环境、多端开发能力以及多端分发机制，如图1-2所示。



图 1-2 “一次开发，多端部署”示意图

1. 多端开发环境

HUAWEI DevEco Studio是面向全场景多设备提供的一站式开发平台，支持多端双向预览、分布式调优、分布式调试、超级终端模拟、低代码可视化开发等能力，帮助开发者降低成本、提升效率、提高质量。

HUAWEI DevEco Studio提供的核心能力如图1-3所示。

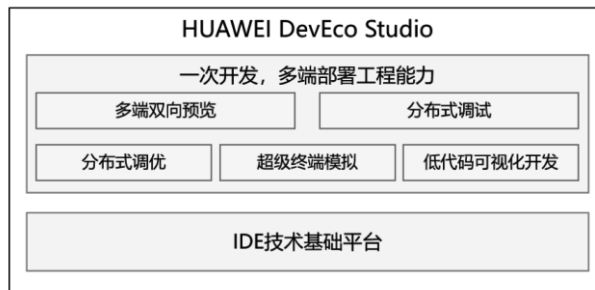


图 1-3 HUAWEI DevEco Studio 提供的核心能力

- **多端双向预览：**在HarmonyOS应用的开发阶段，因不同设备的屏幕分辨率、形状、大小等差异，开发者需要在不同设备上查看UI界面显示，确保实现效果与设计目标一致。在传统的开发模式下，开发者需要获取大量不同的真机设备用于测试验证。HUAWEI DevEco Studio提供了多种设备的双向预览能力，支持同时查看UI代码在多个设备上的预览效果，并支持UI代码和预览效果的双向定位修改。
- **分布式调优：**HarmonyOS应用具有天然的分布式特征，体现在同一个应用在多个设备之间会有大量的交互。开发过程中，对这些交互进行调试时，需要对每个设备分别建立调试会话，并且需要在多个设备之间来回切换，容易造成调试不连续、操作烦琐等问题。为了提升开发效率，HUAWEI DevEco Studio提供了分布式调试功能，支持跨设备调试，通过代码断点和调试堆栈可以方便地跟踪不同设备之间的交互，用于定位多设备互动场景下的代码缺陷。

- **分布式调试**: 分布式应用的运行性能至关重要。在跨端迁移场景中, 需要应用在目标设备上快速启动, 以实现和原设备之间的无缝衔接; 在多端协同场景中, 需要应用在算力和资源不同的多个设备上都能高效运行, 以获得整体的流畅体验。以往开发者在分析分布式应用的性能问题时, 需要单独查看每个设备的性能数据, 并手动关联分析这些数据, 操作烦琐、复杂度高。HUAWEI DevEco Studio提供了分布式调优功能, 支持多设备分布式调用链跟踪、跨设备调用堆栈缝合, 同时采集多设备性能数据并进行联合分析。
- **超级终端模拟**: 移动应用开发时需要使用本地模拟器来进行应用调试, 实现快速开发的目的。HarmonyOS应用需要运行在多种不同类型的设备上, 为此, HUAWEI DevEco Studio提供了不同类型的终端模拟, 支持开发者在多个模拟终端上进行开发调试, 降低门槛、节约成本。同时, 多个模拟终端、真机设备也可以自由地组成超级终端, 进一步降低开发者获取分布式调测环境的难度。
- **低代码可视化开发**: 低代码开发提供UI可视化开发能力, 支持自由拖拽组件和可视化数据绑定, 可快速预览效果, 所见即所得。通过拖拽式编排、可视化配置的方式, 帮助开发者减少重复性的代码编写, 快速地构建多端应用程序。低代码开发的产物如组件、模板等可以被其他模块的代码引用, 并且能通过跨工程复用, 支持开发团队协作完成复杂应用的开发。

2. 多端开发能力

HarmonyOS应用如需在多个设备上运行, 需要适配不同的屏幕尺寸和分辨率、不同的交互方式(如触摸和键盘等)、不同的硬件能力(如内存差异和器件差异等), 开发成本较高。因此, 多端开发能的核心目标是降低多设备应用的开发成本。为了实现该目标, HarmonyOS提供了多端UI适配、交互事件归一、设备能力抽象等核心功能, 帮助开发者降低开发与维护成本, 提升代码复用度。图1-4所示为HarmonyOS对屏幕进行逻辑抽象。

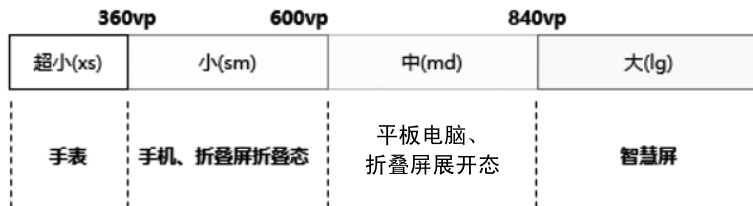


图 1-4 HarmonyOS 对屏幕进行逻辑抽象

- **多端UI适配**: 不同设备屏幕尺寸、分辨率等存在差异, HarmonyOS将对屏幕进行逻辑抽象, 包括尺寸和物理像素, 并提供丰富的自适应/响应式的布局和视觉能力, 方便开发者进行不同屏幕的界面适配。
- **交互事件归一**: 不同设备间的交互方式等存在差异, 如触摸、键盘、鼠标、语音、手写笔等, HarmonyOS将不同设备的输入映射成归一交互事件, 从而简化开发者适配逻辑。
- **设备能力抽象**: 不同设备间的软、硬件能力等存在差异, 如设备是否具备定位能力、是否具备摄像头、是否具备蓝牙功能等, HarmonyOS要对设备能力进行逻辑抽象, 并提供接口来查询设备是否支持某一能力, 方便开发者进行不同软、硬件能力的功能适配。在HarmonyOS中, 使用SystemCapability(简称为 SysCap)定义每个部件对应用开发者提供的系统软、硬件能力。应用开发者基于统一的方式访问不同设备的能力。

3. 多端分发机制

如果需要开发多设备上运行的应用，一般会针对不同类型的设备多次开发并独立上架。开发和维护的成本大，为了解决这个问题，HarmonyOS提供了“一次开发，多端部署”的能力，开发者开发多设备应用，只需要一套工程，一次打包出多个HAP，统一上架，即可根据设备类型按需进行分发，如图1-5所示。

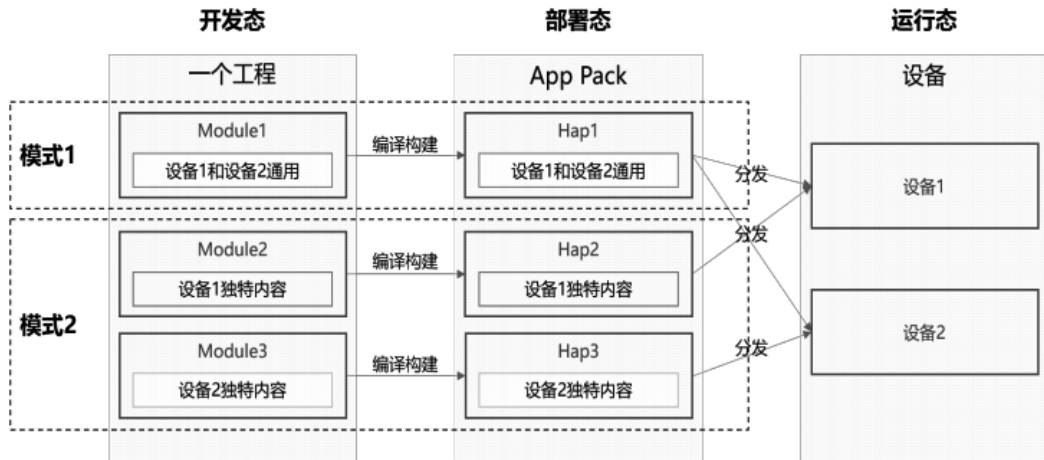


图 1-5 多端分发机制

除了可以开发传统的应用，开发者还可以开发元服务。元服务是一种面向未来的服务提供方式，具有独立入口的、免安装的、可为用户提供一个或多个便捷服务的应用程序形态。HarmonyOS为元服务提供了更多的分发入口，方便用户获取，同时也增加了元服务露出的机会。

1.2.2 可分可合，自由流转

元服务是HarmonyOS提供的一种全新的应用形态，具有独立入口，用户可通过单击、碰一碰、扫一扫等方式直接触发，无须显式安装，由程序框架后台静默安装后即可使用，可为用户提供便捷服务。

传统移动生态下，开发者通常需要开发一个原生应用版本，如果提供小程序给用户，往往需要开发若干个独立的小程序。HarmonyOS生态下，HarmonyOS原生支持元服务开发，开发者无须维护多套版本，而是通过业务解耦将应用分解为若干元服务独立开发，按需根据场景组合成复杂应用。

元服务基于HarmonyOS API开发，支持运行在1+8+N设备上，供用户在合适的场景、合适的设备上便捷使用。元服务是支撑可分可合、自由流转的轻量化程序实体，帮助开发者的服务更快触达用户。具备如下特点：

- 触手可及：元服务可以在服务中心发现并使用，同时也可以基于合适场景被主动推荐给用户使用，例如用户可以在服务中心和小艺建议中发现系统推荐的服务。
- 服务直达：元服务无须安装和卸载，秒开体验，即点即用，即用即走。
- 万能卡片：支持用户无须打开元服务便可获取服务内重要信息的展示和动态变化，如天气、关键事务备忘、热点新闻列表等。
- 自由流转：元服务支持运行在多设备上并按需跨端迁移，或者多个设备协同起来给用户提供最优的体验。

例如手机上未完成的邮件，迁移到平板电脑上继续编辑，手机用作文档翻页和批注，配合智慧屏完成分布式办公；例如分布式游戏场景，手机可作为手柄，与智慧屏配合玩游戏，获得新奇游戏体验。

如图1-6所示是HarmonyOS打包上架模式。

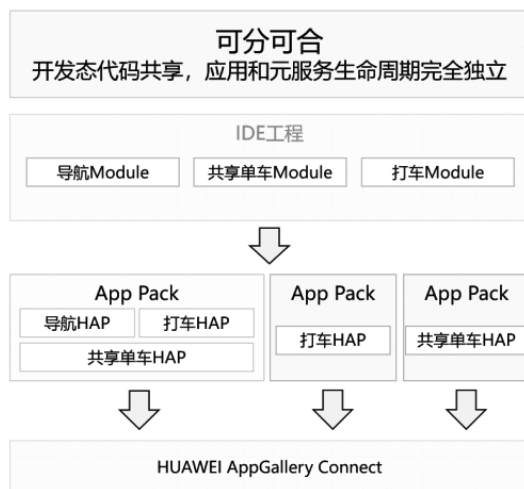


图 1-6 打包上架模式

1.2.3 统一生态，原生智能

1. 统一生态

移动操作系统和桌面操作系统的跨平台应用开发框架存在差异，从渲染方式的角度可以归纳为WebView渲染、原生渲染和自渲染这三类，HarmonyOS提供了系统WebView、ArkUI框架和XComponent能力用于支撑这三种类型的跨平台框架的接入。

主流跨平台开发框架已有版本正在适配HarmonyOS，基于这些框架开发的应用可以以较低成本迁移到HarmonyOS。

2. 原生智能

HarmonyOS内置强大的AI能力，面向HarmonyOS生态应用的开发，通过不同层次的AI能力开放，满足开发者的不同开发场景下的诉求，降低应用的开发门槛，帮助开发者快速实现应用智能化。

1.3 HarmonyOS NEXT 新特性

自HarmonyOS 3.1开始，HarmonyOS推出应用开发Stage模型，并在ArkTS语言、应用程序框架、Web、ArkUI等子系统能力方面有所更新或增强。

HarmonyOS 3.1开放的功能包括：

- Ability框架新增Stage开发模型，包含Stage模型生命周期管理、调度、回调、上下文获取、鉴权等。同时增强了应用的运行管理能力。
- ArkUI开发框架增强了声明式Canvas/XComponent组件的能力，增强了组件布局能力及状态管理能力，优化了部分组件的易用性。

- 应用包管理新增查询应用、Ability和ExtensionAbility相关属性的接口。
- 公共基础类库新增支持Buffer二进制读写。
- Web服务新增支持文档类Web应用的文档预览和基础编辑功能，以及cookie的管理和存储管理。
- 图形图像新增支持YUV、webp图片编解码等能力；新增native vsync能力，支持自绘制引擎自主控制渲染节奏。
- 媒体服务新增相机配置与预览功能。
- 窗口服务新增Stage模型下窗口相关接口，增强窗口旋转能力，增强避让区域查询能力。
- 全球化服务新增支持时区列表、音译、电话号码归属地等国际化增强能力。
- 公共事件基础能力增强，commonEvent模块变更为commonEventManager。
- 资源管理服务新增资源获取的同步接口，新增基于名称查询资源值的接口，新增number、float资源类型查询接口，新增Stage模型资源查询方式。
- 输入法服务新增输入法光标方向常量。

从HarmonyOS NEXT Developer Preview1（API 11）版本开始，HarmonyOS SDK以Kit维度提供丰富、完备的开放能力，涵盖应用框架、系统、媒体、图形、应用服务、AI 6大领域：

- 应用框架相关Kit开放能力：Ability Kit（程序框架服务）、ArkUI（方舟UI框架）等。
- 系统相关Kit开放能力：Universal Keystore Kit（密钥管理服务）、Network Kit（网络服务）等。
- 媒体相关Kit开放能力：Audio Kit（音频服务）、Media Library Kit（媒体文件管理服务）等。
- 图形相关Kit开放能力：ArkGraphics 2D（方舟2D图形服务）、Graphics Accelerate Kit（图形加速服务）等。
- 应用服务相关Kit开放能力：Game Service Kit（游戏服务）、Location Kit（位置服务）等。
- AI相关Kit开放能力：Intents Kit（意图框架服务）、HiAI Foundation Kit（HiAI Foundation服务）等。

1.3.1 Stage 模型

从API 9开始，Ability框架引入并支持使用Stage模型进行开发。因此，Ability框架模型结构具有两种形态：

- FA模型：API 8及其更早版本的应用程序只能使用FA模型进行开发。
- Stage模型：从API 9开始，Stage模型只支持使用ArkTS语言进行开发。

Stage模型是HarmonyOS 3.1开始新增的模型，也是目前HarmonyOS主推且将长期演进的模型。在该模型中，由于提供了AbilityStage、WindowStage等类作为应用组件和Window窗口的“舞台”，因此称这种应用模型为Stage模型。本书主要介绍以Stage模型为主的开发方式。

1.3.2 Ability 组件的生命周期

Ability组件的生命周期切换以及和AbilityStage、WindowStage之间的调度关系如图1-7所示。

Stage模型定义Ability组件的生命周期，只包含创建、销毁、前台和后台等状态，而将与界面强相关的获焦、失焦状态都放在WindowStage中，从而实现Ability与窗口之间的弱耦合；在服务侧，窗口管理服务依赖于组件管理服务，前者通知后者前台和后台变化，这样组件管理服务仅感知前台和后台变化，不感知焦点变化。

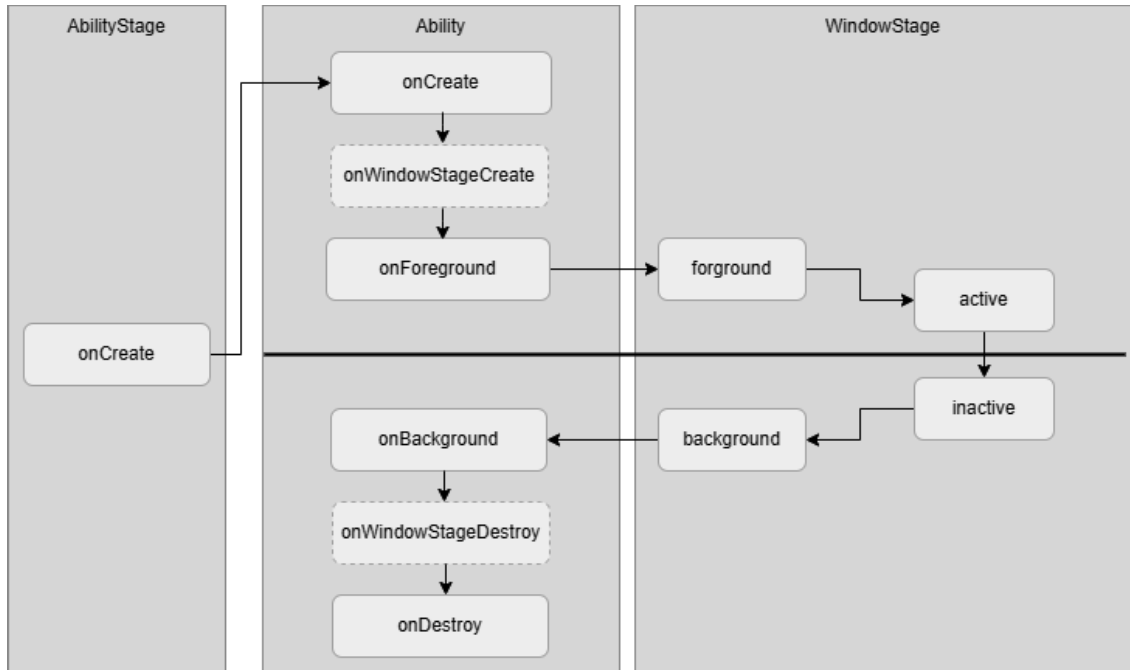


图 1-7 Ability 组件的生命周期

需要注意的是，在 Ability 中存在两个与 WindowStage 相关的生命周期状态：onWindowStageCreate 和 onWindowStageDestroy，这两个生命周期状态的变化仅存在于具有显示能力的设备中。前者表示 WindowStage 已经创建完成，开发者可以通过执行 loadContent 的操作设置 Ability 需要加载的页面；后者在 WindowStage 销毁后调用，以便开发者将资源进行释放。

1.3.3 ArkUI 开发框架

ArkUI（方舟 UI 框架）是一个具有简洁、高性能、支持跨设备的 UI 框架，它提供了丰富的应用界面开发所需的能力。

ArkUI 包括 UI 组件、动画、绘制、交互事件、JS API 扩展机制等。

在 HarmonyOS NEXT 版本，ArkUI 新增或增强了以下能力：

- ArkUI 针对三方框架场景提供组件 NDK 接口，涵盖组件创建、属性设置、事件注册、自定义能力、组件树构建。
- 自定义节点能力增强，提供 FrameNode 的自定义能力以及节点代理能力，并提供基础组件、手势、事件的 Modifier 能力。
- Navigation 新增页面生命周期：支持 onWillShow 和 onWillHide 页面生命周期、转场动画支持打断和接续、页面内容扩展到状态栏、单例跳转能力和自定义动效能力增强。
- Video/XComponent/Canvas 组件支持 AI 能力，支持文本和实体识别。Text 等文本类组件支持属性字符串。
- 支持使用 NativeWindowEventFilter 能力拦截按键事件。
- 支持使用画中画功能。
- 新增提供智慧多窗的适配指导。

1.3.4 ArkTS 编程语言

ArkTS（方舟编程语言）提供HarmonyOS应用开发语言ArkTS相关的公共核心基础能力接口，包括并发、容器、流、文本编解码、XML、URI、Buffer等能力。

ArkTS是HarmonyOS优选的主力应用开发语言。ArkTS基于TypeScript（简称TS）语言扩展而来，是TS的超集（这也是为什么ArkTS的原名叫eTS，是extend TypeScript的简写）。ArkTS继承了TS的所有特性，并且ArkTS在TS基础上还扩展了声明式UI能力，让开发者以更简洁、更自然的方式开发高性能应用。

在HarmonyOS NEXT版本，ArkTS新增或增强了以下能力：

- Taskpool新增支持任务组、串行队列、长时任务、取消任务、宿主通信、设置任务监听、设置任务依赖关系，新增受限worker能力。
- 新增流基础能力，包括流读、写、双工和转换。
- 增强Uri、TextDecoder、StringDecoder等模块能力。
- Taskpool、TextDecoder、buffer、uri等模块接口性能优化。

1.3.5 ArkWeb

ArkWeb（方舟Web）提供了在应用中使用Web页面的能力，支持应用集成Web页面、小程序、浏览器网页浏览等场景的混合开发。

在HarmonyOS NEXT版本，ArkWeb新增或增强了以下能力：

- 网页加载与页面导航：新增支持打开UniversalLink链接、应用托管网络、应用级自定义DNS。
- 网页渲染与显示：新增支持object/embed标签的同层渲染与事件传递、长网页渲染模式、网页长截图、扩展安全区域与H5避让区查询能力、预览PDF（内置PDFView扩展）。
- UX一致性增强：增强支持文本选择智能选词、图片长按识文、列表滑动曲线与原生一致。
- 网页媒体：新增支持显示HEIF图片、网页视频托管、网页音视频与摄像头控制。
- 网页安全隐私：新增广告过滤能力支持自定义拦截规则，智能防跟踪及限制三方Cookie的访问，网页高级安全模式。
- W3C兼容性增强：支持设置meta标签的viewport属性，支持鼠标悬停提示tooltip、datalist元素，支持自定义鼠标指针样式等。
- 性能增强：新增ArkWeb组件动态创建与上下树、JavaScript接口性能提升、V8引擎性能优化。
- DFX增强：新增了对独立Web GPU进程的支持，以增强应用的稳定性和安全性。应用现在可以启用独立的Web渲染进程，这不仅提高了应用的运行安全性，还增强了稳定性。同时，我们优化了无障碍常用功能，并且增强了网页性能度量接口（LCP/FMP），以便更准确地监测性能。此外，我们还集成了crashpad崩溃信息生成功能，以便于崩溃问题的分析和修复。

1.3.6 “纯血鸿蒙”解读

2023年8月4日，在华为开发者大会上，华为发布HarmonyOS NEXT开发者预览版本。据介绍，HarmonyOS NEXT系统底座全线自研，去掉了传统的AOSP代码，不再兼容安卓开源应用，仅支持鸿蒙内核和系统的应用，因此也被称为“纯血鸿蒙”。

HarmonyOS NEXT可以理解HarmonyOS面向未来的、自研程度更高的下一代鸿蒙系统。

“纯血鸿蒙”具有以下特点：

- 全面自研：鸿蒙系统通过全新的架构和核心技术实现了全面自研，正式脱离了Android的影响，成为真正独立的操作系统。
- 自主可控：鸿蒙内核、文件系统、编程语言和编译器等均为华为自主研发，确保了系统的安全性和稳定性。
- 高度弹性：鸿蒙内核具有高度弹性的架构，能够根据硬件需求灵活组合操作系统能力，满足各种终端的需求。
- 鸿蒙内核：作为“纯血鸿蒙”的核心，鸿蒙内核提供了更安全、更流畅、更弹性的系统基础。其服务之间能够更好地隔离，从架构上保证了系统的安全性。
- 文件系统：使用自研的文件系统，进一步提升了系统的稳定性和性能。
- 编程语言与编译器：鸿蒙还自主研发了编程语言和编译器，确保了开发过程的独立性和高效性。
- 系统隔离：鸿蒙内核的服务之间隔离性更好，从架构上保证了系统的安全性，获得了全球首张智能终端领域CC EAL 6+证书。
- 隐私保护：新的隐私保护机制极大地减少了权限弹窗，提升了用户的隐私安全。

综上所述，“纯血鸿蒙”是华为在操作系统领域的一次重大突破和创新成果。它不仅代表了华为在自研技术方面的实力和决心，也为全球科技界带来了新的惊喜和期待。

1.4 DevEco Studio 的安装

要想快速体验HarmonyOS应用开发，IDE必不可少，而DevEco Studio是华为官方指定的HarmonyOS集成开发环境。所谓“工欲善其事，必先利其器”，本节将介绍DevEco Studio的安装步骤。

1.4.1 下载 DevEco Studio

目前，HarmonyOS专属IDE的DevEco Studio开发环境，可以从HarmonyOS官网免费下载使用。下载地址为<https://developer.huawei.com/consumer/cn/download/>。

DevEco Studio支持Windows（64-bit）、Mac（X86）、Mac（ARM）3个操作系统平台。

下面以Windows（64-bit）操作系统为例，下载并获得devecostudio-windows-x.x.x.xxx.zip压缩包，其中“x.x.x.xxx”为实际下载的版本号。解压该压缩包，就能得到一个deveco-studio-x.x.x.xxx.exe安装文件。

1.4.2 安装 DevEco Studio

双击deveco-studio-x.x.x.xxx.exe文件进行安装，进入DevEco Studio安装向导。在如图1-8所示的界面选择安装路径，默认安装于C:\Program Files路径下，也可以单击“浏览（B）...”指定其他安装路径，然后单击“下一步”按钮。

在如图1-9所示的安装选项界面中勾选“创建桌面快捷方式”后，单击“下一步”按钮，直至安装完成。



图 1-8 选择安装位置



图 1-9 勾选创建桌面快捷方式

看到操作系统桌面有如图1-10所示的快捷方式，则证明安装已经完成。

1.4.3 配置 DevEco Studio

双击DevEco Studio桌面快捷方式启动DevEco Studio。

如果之前使用过DevEco Studio并有保存DevEco Studio的配置，则可以导入DevEco Studio的配置；否则，选择“Do not import settings”选项，单击OK按钮，如图1-11所示。

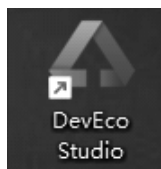


图 1-10 快捷方式

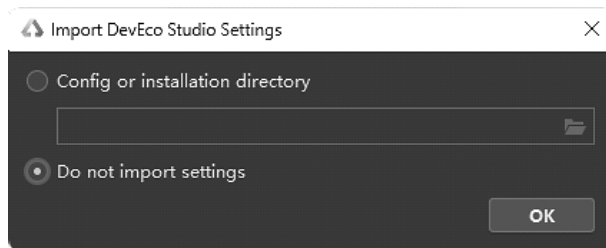


图 1-11 选择“Do not import settings”选项

首次使用DevEco Studio会弹出如下提示信息，单击“Agree”按钮继续执行下一步，如图1-12所示。

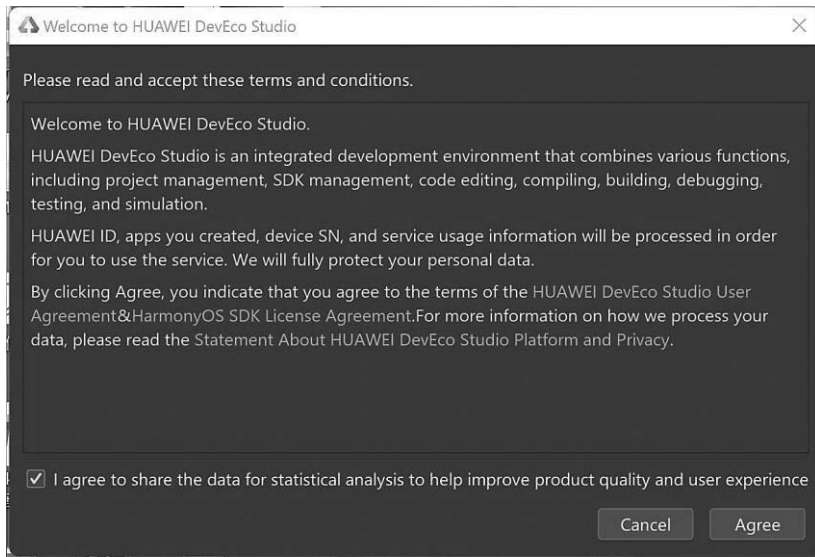


图 1-12 单击“Agree”按钮

此时会进入欢迎界面。在“More Actions”下拉选项中选择“Device Manager”，如图1-13所示。

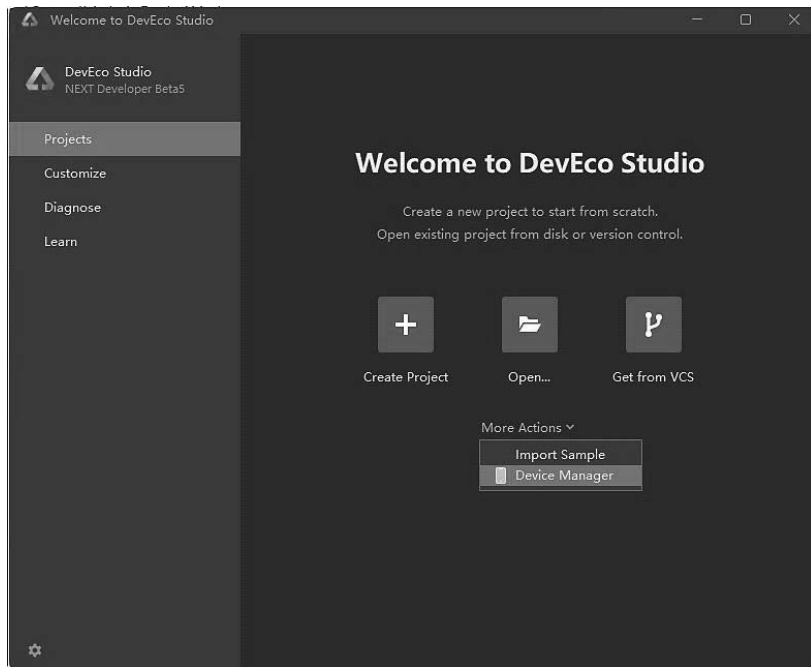


图 1-13 选择“Device Manager”

进入Device Manager（设备管理器）页面之后，就可以创建虚拟机了。

1.4.4 创建虚拟机

进入Device Manager（设备管理器）页面之后，可以看到本地模拟器列表页面，如图1-14所示。

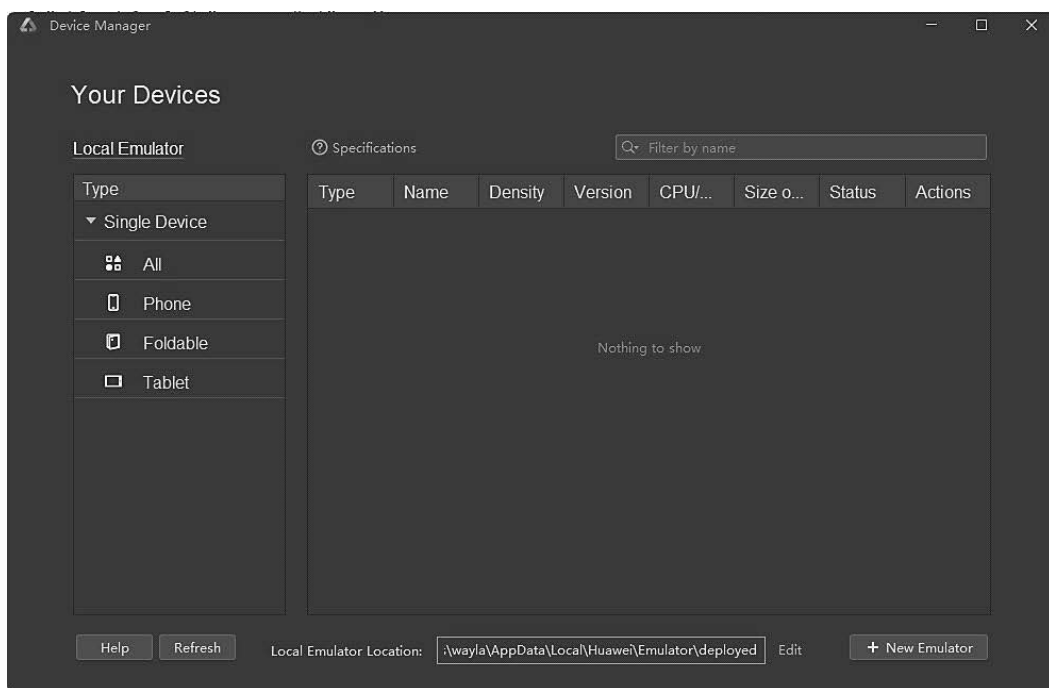


图 1-14 本地模拟器列表页面

单击“New Emulator”按钮创建虚拟机，弹出如图1-15所示的虚拟机镜像页面。

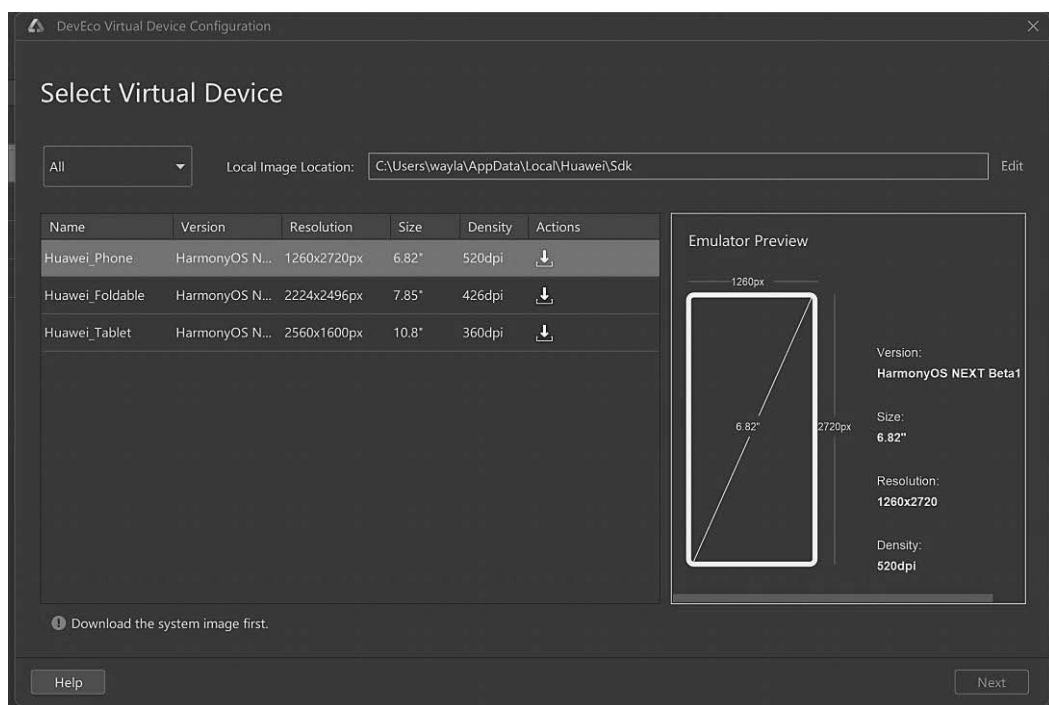


图 1-15 虚拟机镜像页面

如果之前没有下载过虚拟机镜像，则需要先单击如图1-16所示的Actions中的下载按钮下载虚拟机镜像。

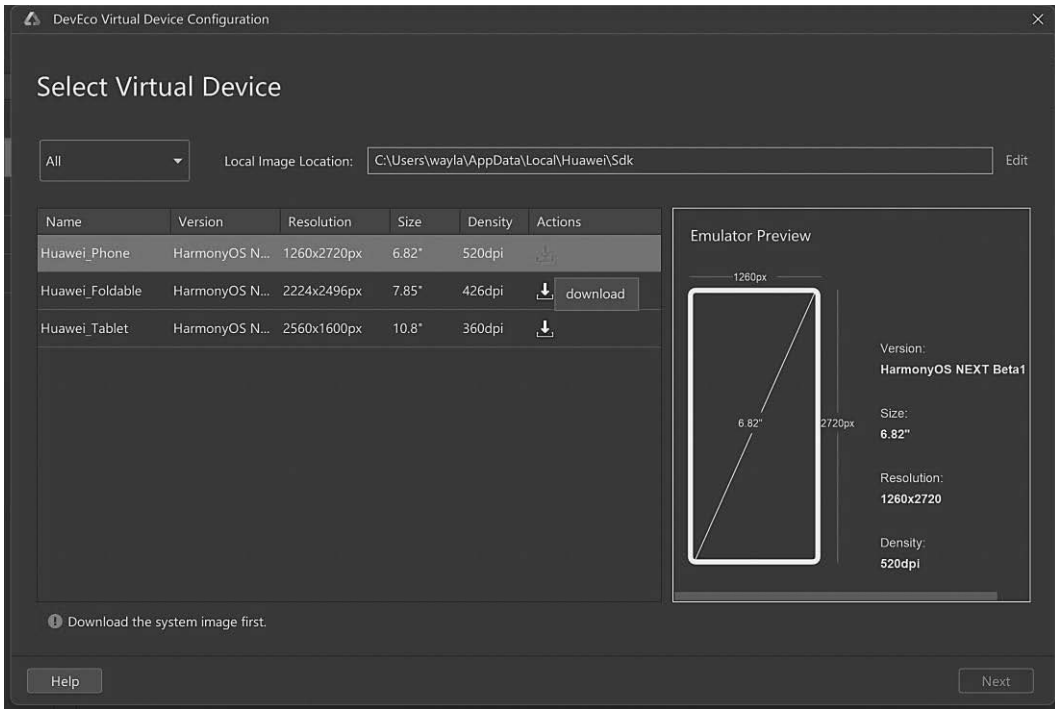


图 1-16 单击下载虚拟机镜像

弹出License Agreement对话框，单击“Accept”单选按钮，之后单击Next按钮，如图1-17所示。

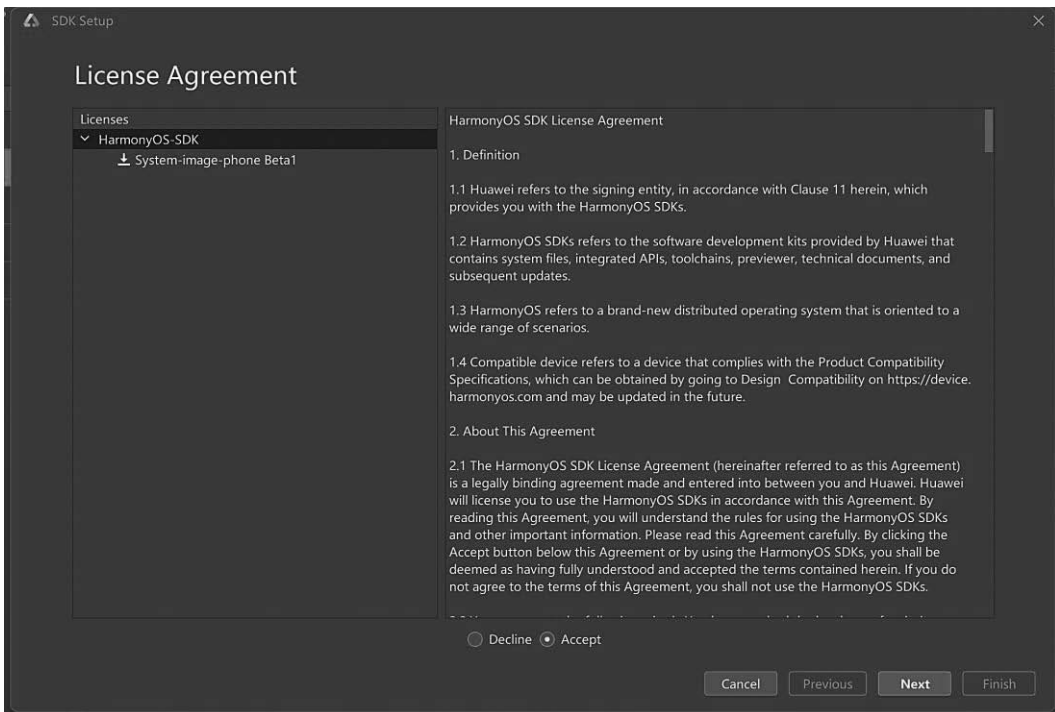


图 1-17 单击“Accept”单选按钮

如果一切顺利，将看到如图1-18所示的页面，则证明虚拟机镜像下载完成。

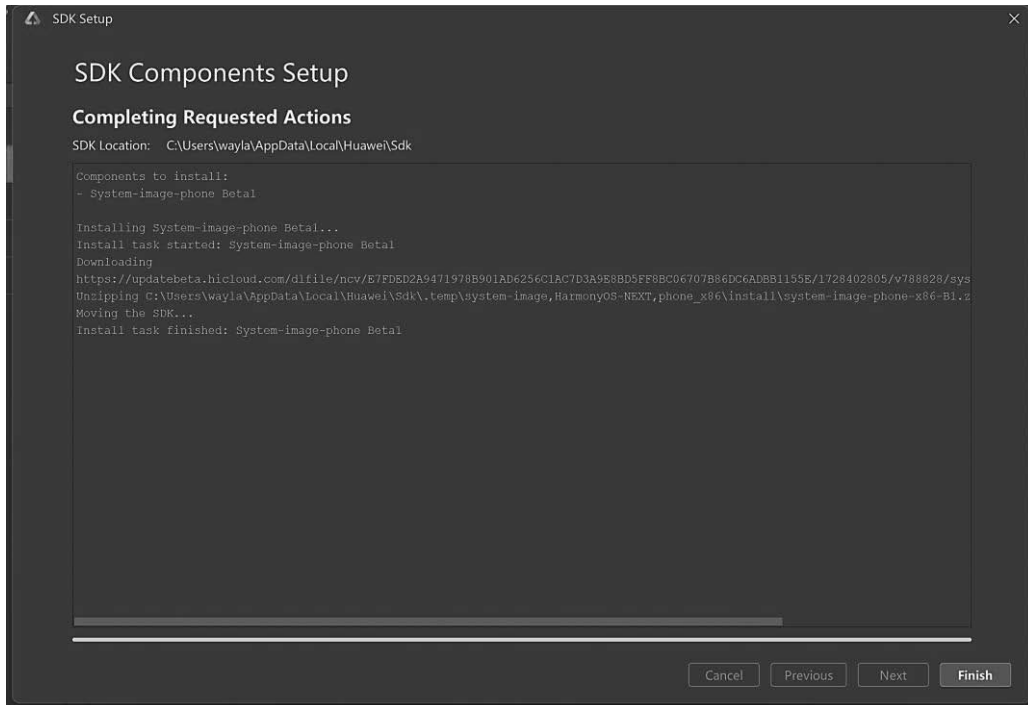


图 1-18 虚拟机镜像下载完成

当虚拟机镜像下载完成之后，则可创建虚拟机，如图1-19所示。

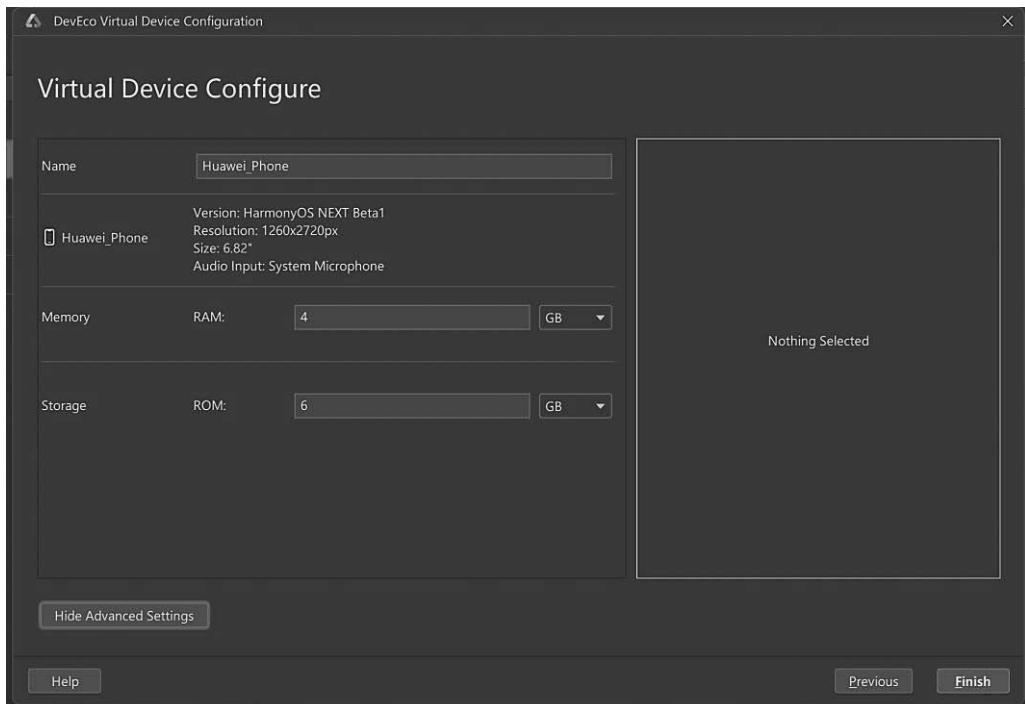


图 1-19 创建虚拟机

虚拟机创建完成后就会出现在本地模拟器列表页面中，如图1-20所示。

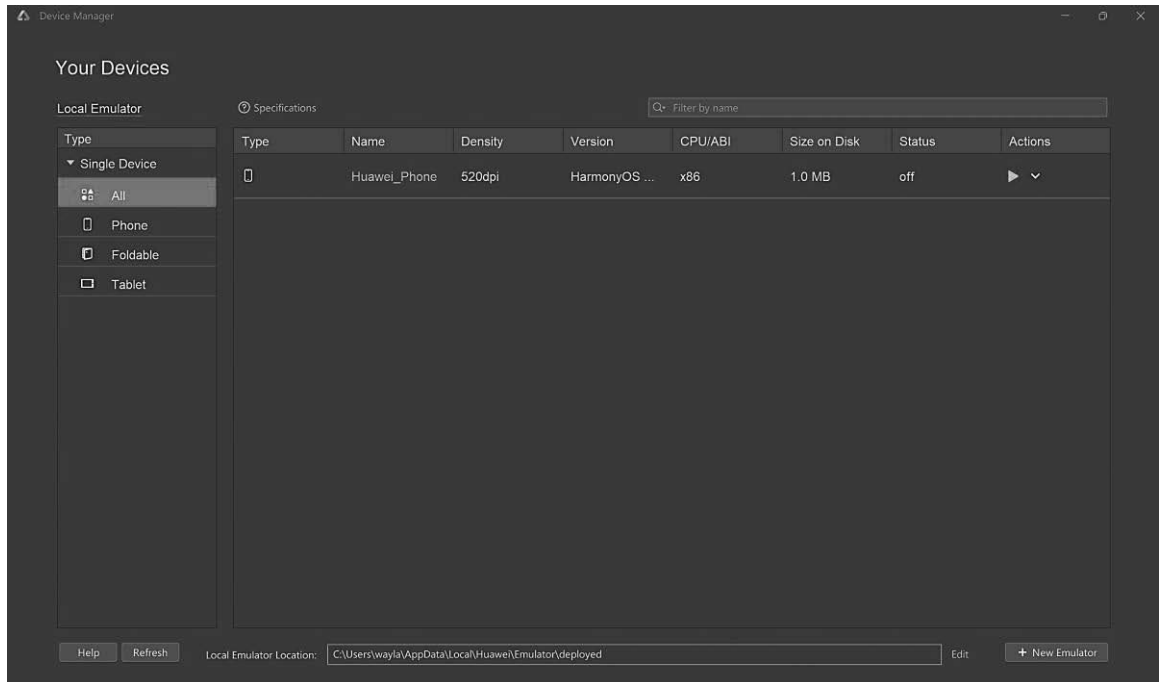


图 1-20 虚拟机创建完成

在创建的虚拟机上单击启动按钮，则可启动虚拟机。虚拟机的启动效果如图1-21所示。

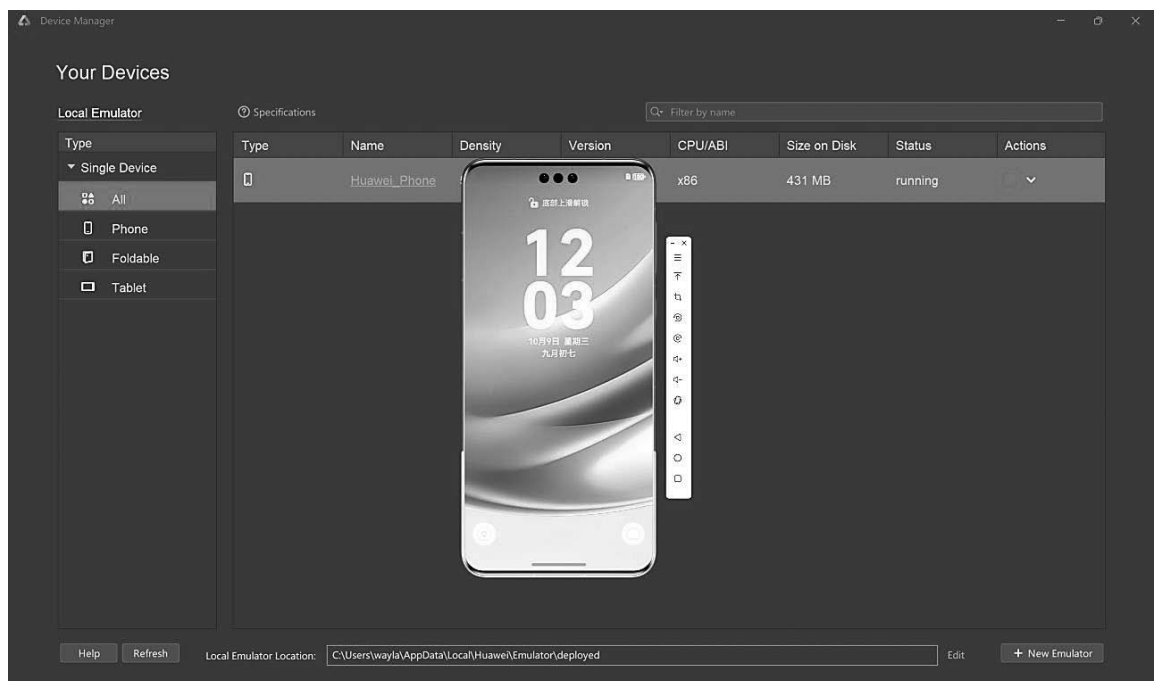


图 1-21 虚拟机的启动效果

以上就是创建Phone类型虚拟机的过程。其他类型的设备，比如Foldable、Tablet等，其虚拟机的创建过程也类似。

1.5 实战：创建第一个 HarmonyOS NEXT 应用

本节我们将演示基于DevEco Studio来开发第一个基于HarmonyOS NEXT版本的应用。

1.5.1 选择创建新项目

在打开DevEco Studio后，可以看到如图1-22所示的欢迎界面。我们单击“Create Project”来创建一个新项目。

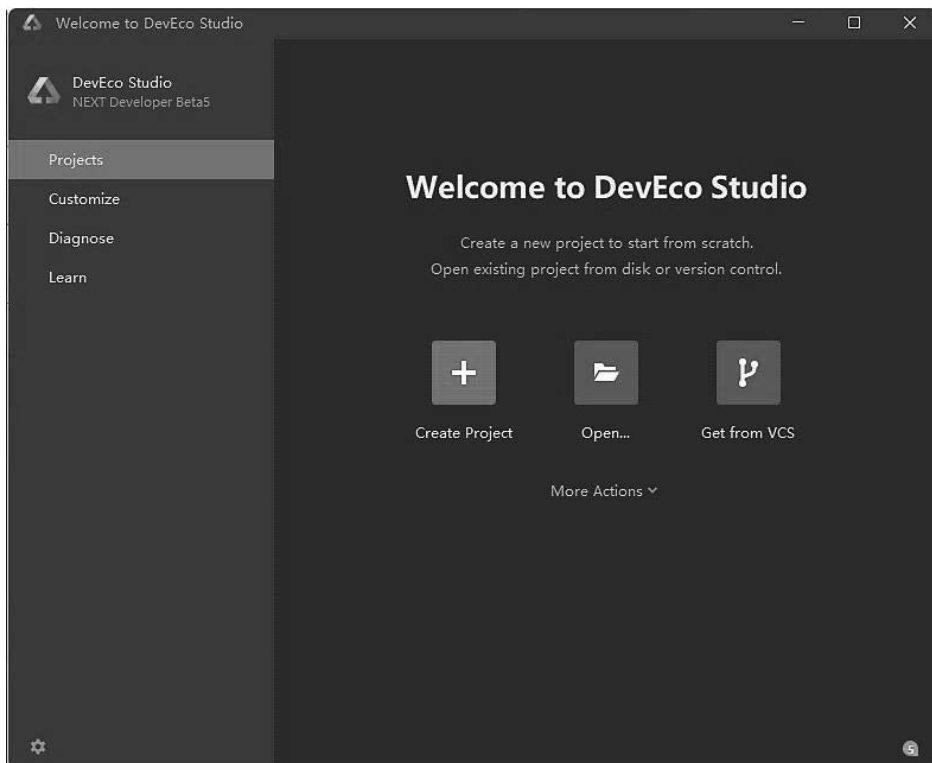


图 1-22 创建一个新项目

后续如果是在已打开项目的状态下，也可以从DevEco Studio菜单选择“File→New→Create Project”来创建一个新项目。

1.5.2 选择模板

在新项目对话框中许多不同应用类型的工程模板，如图1-23所示。

DevEco Studio支持多种类型的应用/服务开发，预置丰富的工程模板，可以根据工程向导轻松创建适应于各类设备的工程，并自动生成对应的代码和资源模板。同时，DevEco Studio还提供了多种编程语言供开发者进行应用/服务开发，包括ArkTS、JS和C/C++等。

本例所选择的“Empty Ability”的应用工程模板，是支持Phone、Tablet、2in1、Car设备的模板，用于展示基础的Hello World功能。有关Ability的概念，我们后续再进行介绍。

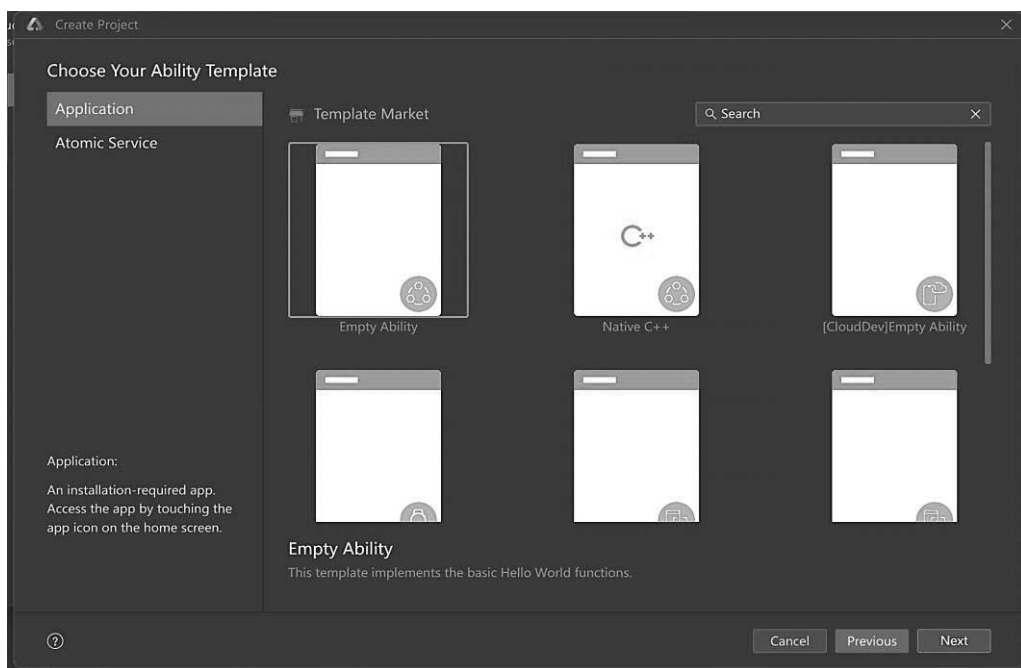


图 1-23 选择工程模板

1.5.3 配置项目信息

下面需要配置项目的信息，比如项目名称、包名、位置、SDK版本等，按照个人实际情况填写，如图1-24所示。

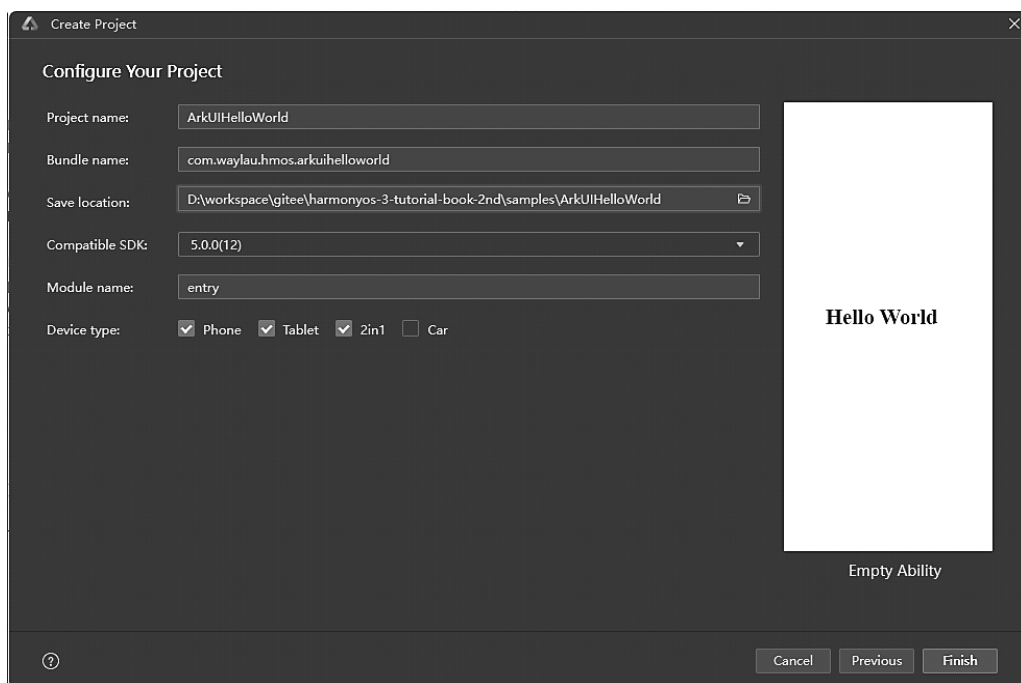


图 1-24 配置项目的信息

这些配置详细信息如下：

- **Project name:** 表示开发者可以自行设置项目名称，此处可以修改为自己的项目名称。本例命名为“ArkUIHelloWorld”。
- **Bundle name:** 表示包名称，默认情况下，应用ID也会使用该名称，在应用发布时对应的ID需要保持一致。
- **Save location:** 表示工程保存路径，建议用户自行设置相应位置。由大小写字母、数字和下划线等组成，不能包含中文字符。
- **Compatible SDK:** 表示应用能兼容的最低API版本。
- **Device type:** 表示该工程模板支持的设备类型。本例选用Phone、Tablet、2in1作为设备类型。

1.5.4 自动生成工程代码

单击“Finish”按钮之后，DevEco Studio就为我们创建好了整个应用，并且会自动生成工程代码，对工程进行构建，如图1-25所示。

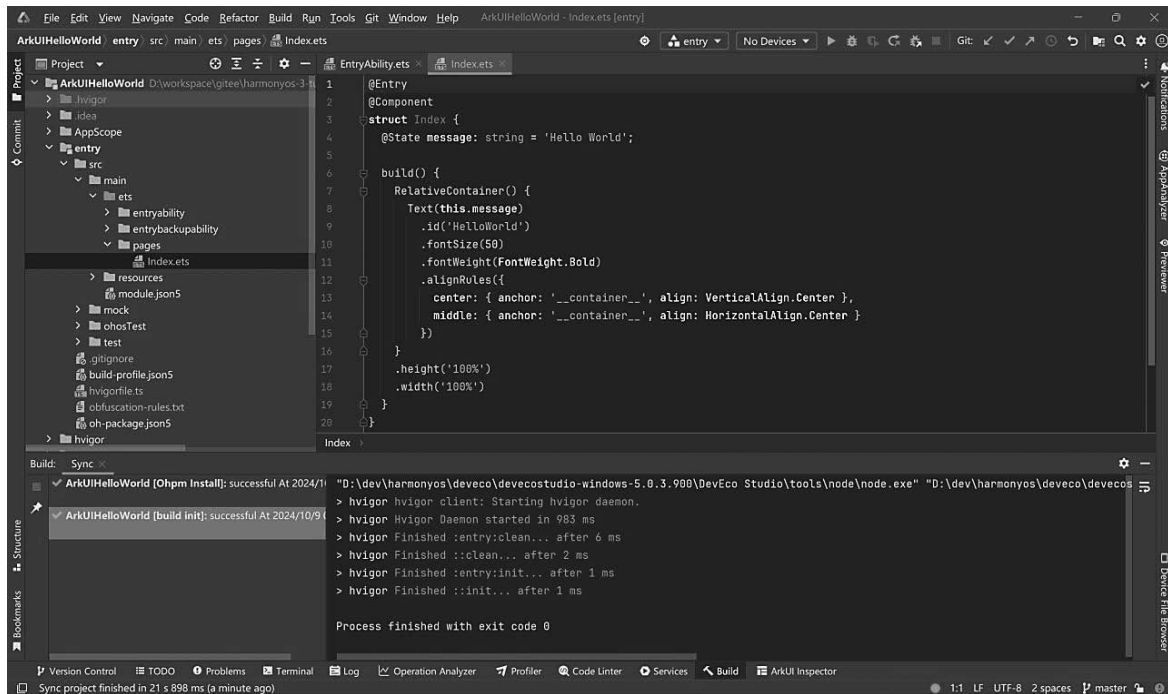


图 1-25 自动生成工程代码

将上述工程代码初始化完成之后，就能在该程序基础上进行代码开发、运行了。

1.5.5 预览项目

我们可以使用预览器来预览项目。打开预览器有两种方式：

- 在菜单栏中单击“View→Tool Windows→Previewer”，打开预览器。
- 在编辑窗口右上角的侧边工具栏，单击“Previewer”按钮，打开预览器。

预览器的显示效果如图1-26所示。

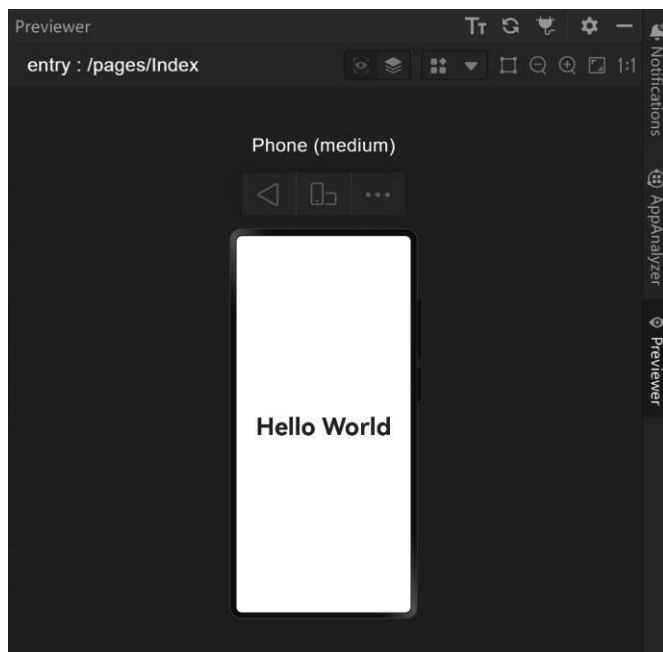


图 1-26 预览器显示效果

1.5.6 运行项目

HarmonyOS支持包括模拟器、真机等多种方式来运行项目。

上述方式各有利弊，比如真机是需要读者自己准备具有HarmonyOS系统的真实设备（比如手机、平板电脑），且该系统版本需要升级至最新版本；与真机相比，模拟器会缺失部分真机的功能，比如拍照、指纹等。模拟器与真机的差异详见“附录”。

注意 本书采用模拟器方式来运行项目，本书内容也尽量规避无法用模拟器演示的功能。

在菜单栏中单击“Tools→Device Manager”进入设备管理界面，在设备模拟器列表中，选择要启动的模拟器（以Phone设备为例）。如图1-27所示，在已启动的模拟器列表中选择要使用的模拟器，并单击三角形按钮以启动项目。

在模拟器中项目运行的效果如图1-28所示。

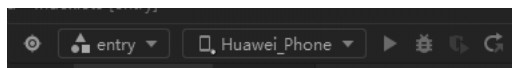


图 1-27 启动项目



图 1-28 项目运行效果

以上就是运行项目的完整过程。

1.6 应用工程结构介绍

本节介绍应用工程结构及各个配置文件的含义。

1.6.1 App 包结构

在进行应用/服务开发前，开发者应该掌握应用/服务的逻辑结构。

应用/服务发布形态为App Pack（Application Package），它是由一个或多个HAP（Harmony Ability Package）包以及描述App Pack属性的pack.info文件组成。

一个HAP在工程目录中对应一个Module，它是由代码、资源、第三方库及应用/服务配置文件组成，HAP可以分为Entry和Feature两种类型。

- Entry: 应用的主模块，作为应用的入口，提供了应用的基础功能。
- Feature: 应用的动态特性模块，作为应用能力的扩展，可以根据用户的需求和设备类型进行选择性安装。

Stage模型的应用程序包结构如图1-29所示。

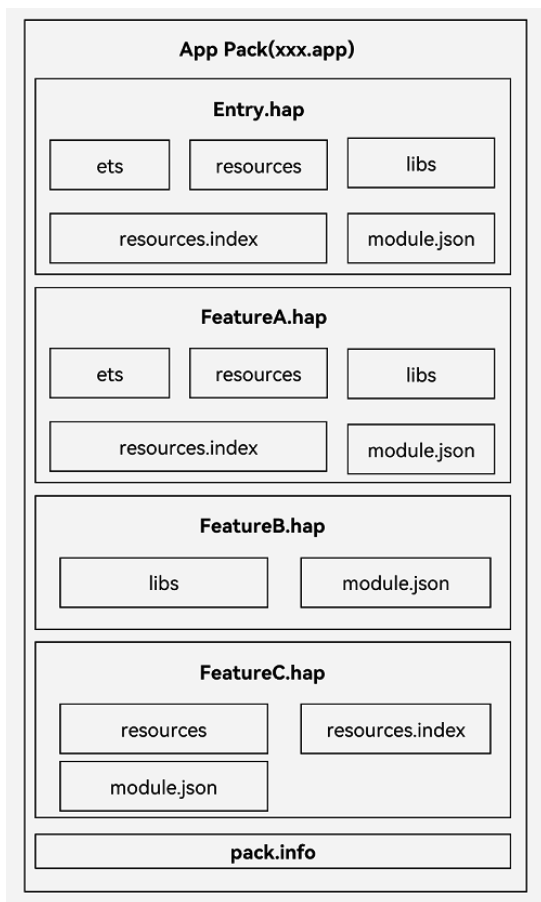


图 1-29 应用程序包结构

1.6.2 工程级目录

下面以“ArkUIHelloWorld”应用为例，ArkTS Stage模型的工程目录结构如图1-30所示。

- AppScope/app.json5: 应用的全局配置信息。
- entry: 应用/服务模块，编译构建生成一个HAP。
- oh_modules: 用于存放第三方库依赖信息，包含应用/服务所依赖的第三方库文件。
- build-profile.json5: 应用级配置信息，包括签名、产品配置等。
- hvmigorfile.ts: 应用级编译构建任务脚本。
- oh-package.json5: 描述全局配置，如依赖覆盖（overrides）、依赖关系重写（overrideDependencyMap）和参数化配置（parameterFile）等。

1.6.3 entry 模块级目录

entry模块级目录含义描述如下：

- src/main/ets: 用于存放ArkTS源码。
- src/main/ets/entryability: 应用/服务的入口。
- src/main/ets/pages: 应用/服务包含的页面。
- src/main/resources: 用于存放应用/服务模块所用到的资源文件，如图形、多媒体、字符串、布局文件等。关于资源文件的详细说明请参考资源分类与访问。
 - ♦ base/element: 包括字符串、整数型、颜色、样式等资源的JSON文件。每个资源均由JSON格式进行定义，例如boolean.json、color.json、string.json等。
 - ♦ base/media: 多媒体文件，如图形、视频、音频等文件，支持的文件格式包括.png、.gif、.mp3、.mp4等。
 - ♦ rawfile: 用于存储任意格式的原始资源文件。rawfile不会根据设备的状态去匹配不同的资源，需要指定文件路径和文件名进行引用。
- src/main/module.json5: Stage模型模块配置文件，主要包含HAP的配置信息、应用在具体设备上的配置信息以及应用的全局配置信息。具体请参考module.json5配置文件。
- build-profile.json5: 当前的模块信息、编译信息配置项，包括buildOption、targets配置等。
- hvmigorfile.ts: 模块级编译构建任务脚本。
- oh-package.json5: 描述三方包的包名、版本、入口文件（类型声明文件）和依赖项等信息。

1.6.4 配置文件

1. app.json5

AppScope/app.json5是应用的全局的配置文件，用于存放应用公共的配置信息。

```
{
  "app": {
```

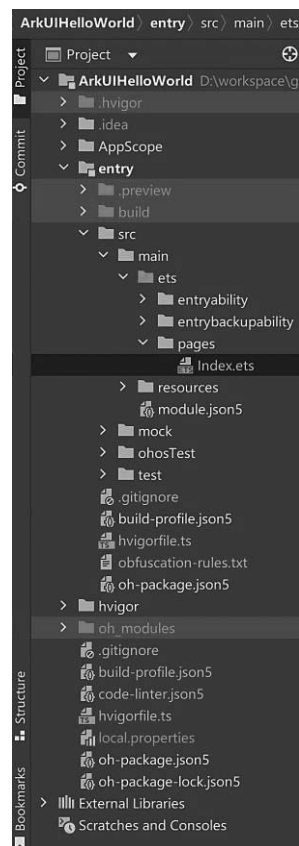


图 1-30 工程目录结构

```
"bundleName": "com.waylau.hmos.arkuihelloworld",
"vendor": "example",
"versionCode": 1000000,
"versionName": "1.0.0",
"icon": "$media:app_icon",
"label": "$string:app_name"
}
}
```

其中配置信息如下：

- `bundleName`是包名。
- `vendor`是应用程序供应商。
- `versionCode`是用于区分应用版本。
- `versionName`是版本号。
- `icon`是对于应用的显示图标。
- `label`是应用名。

2. module.json5

`entry/src/main/module.json5`是模块的配置文件，包含当前模块的配置信息。

```
{
  "module": {
    "name": "entry",
    "type": "entry",
    "description": "$string:module_desc",
    "mainElement": "EntryAbility",
    "deviceTypes": [
      "phone",
      "tablet",
      "2in1"
    ],
    "deliveryWithInstall": true,
    "installationFree": false,
    "pages": "$profile:main_pages",
    "abilities": [
      {
        "name": "EntryAbility",
        "srcEntry": "./ets/entryability/EntryAbility.ets",
        "description": "$string:EntryAbility_desc",
        "icon": "$media:layered_image",
        "label": "$string:EntryAbility_label",
        "startWindowIcon": "$media:startIcon",
        "startWindowBackground": "$color:start_window_background",
        "exported": true,
        "skills": [
          {
            "entities": [
              "entity.system.home"
            ],
            "actions": [
              "action.system.home"
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "extensionAbilities": [
      {
        "name": "EntryBackupAbility",
        "srcEntry": "./ets/entrybackupability/EntryBackupAbility.ets",
        "type": "backup",
        "exported": false,
        "metadata": [
          {
            "name": "ohos.extension.backup",
            "resource": "$profile:backup_config"
          }
        ]
      }
    ]
  }
}

```

其中module对应的是模块的配置信息，一个模块对应一个打包后的HAP包，HAP包全称是HarmonyOS Ability Package，其中包含了Ability、第三方库、资源和配置文件。其具体属性及其描述可以参照表1-1。

表1-1 module.json5默认配置属性及描述

属 性	描 述
name	该标签标识当前module的名字，module打包成HAP后，表示HAP的名称，标签值采用字符串表示（最大长度为31个字节），该名称在整个应用中要唯一
type	表示模块的类型，共有4种，分别是entry、feature、har和shared
description	当前模块的描述信息
mainElement	该标签标识HAP的入口ability名称或者extension名称。只有配置为mainElement的ability或者extension才允许在服务中心露出
deviceTypes	该标签标识HAP可以运行在哪类设备上，标签值采用字符串数组表示
deliveryWithInstall	该标签标识当前HAP是否在用户主动安装时安装，true表示主动安装时安装，false表示主动安装时不安装
installationFree	表示当前HAP是否支持免安装特性，如果此配置项为true，包名必须加上.hservice后缀
pages	对应的是main_pages.json文件，用于配置ability中用到的page信息
abilities	是一个数组，存放当前模块中所有的ability元能力的配置信息，其中可以有多个ability

对于abilities中每一个ability的属性项，其描述信息如表1-2所示。

表1-2 abilities中对象的默认配置属性及描述

属 性	描 述
name	该标签标识当前ability的逻辑名，该名称在整个应用要唯一，标签值采用字符串表示（最大长度为127个字节）
srcEntry	ability的入口代码路径
description	ability的描述信息
icon	ability的图标。该标签标识ability图标，标签值为资源文件的索引。该标签的默认值为空。如果ability被配置为MainElement，该标签也必须配置
label	ability的标签名

(续表)

属 性	描 述
startWindowIcon	启动页面的图标
startWindowBackground	启动页面的背景色
recoverable	标识当前UIAbility组件是否支持在检测到应用故障后，恢复到应用原界面
continueType	标识当前UIAbility组件的跨端迁移类型

3. main_pages.json

src/main/resources/base/profile/main_pages.json文件保存的是页面page的路径配置信息，所有需要进行路由跳转的page页面都要在这里进行配置。

1.7 本章小结

本章主要介绍了HarmonyOS的概念、背景、特征，以及如何通过DevEco Studio来创建HarmonyOS项目。本章也详细介绍了HarmonyOS应用工程结构的含义。这些都是开发HarmonyOS项目的基础，希望读者能够掌握。

1.8 上机练习：开发第一个 HarmonyOS NEXT 应用 ——Hello World

任务要求：安装DevEco Studio、搭建HarmonyOS开发环境，并开发第一个HarmonyOS应用——Hello World。

练习步骤：

- (1) 安装DevEco Studio、搭建HarmonyOS开发环境。
- (2) 开发“Hello World”应用。
- (3) 运行“Hello World”应用。

代码参考配书资源中的“ArkUIHelloWorld”应用。