第 1 章

HarmonyOS介绍

鸿蒙操作系统（HarmonyOS）是一款面向物联网（Internet of Things，IoT）全场景的分布式操作系统。不同于现有的Android、iOS、Windows、Linux等操作系统，鸿蒙操作系统设计的初衷是解决在5G万物互联时代，各个系统间的连接问题。鸿蒙操作系统面向的是1+8+N的全场景设备，能够根据不同内存级别的设备进行弹性组装和适配，可实现跨硬件设备间的信息交互。

1.1 HarmonyOS的发展历程

“鸿蒙”名字源于华为公司内部一个研究操作系统内核的项目代号。“鸿蒙操作系统”的英文名称是HarmonyOS，Harmony意为和谐，可引申为世界大同、和合共生，这是中华文明一直秉持的理念。鸿蒙有盘古开天辟地之意，“鸿蒙初辟原无性，打破顽空须悟空”，鸿蒙生态刚刚起步，需要华为以及其他国内外企业共同努力，需要众多的“悟空”共同推动，构建更加绚丽多彩的鸿蒙生态世界。华为的鸿蒙，中国的鸿蒙，必将成为世界的鸿蒙。鸿蒙操作系统Logo如图1-1所示。

图1-1　鸿蒙操作系统Logo

早在2012年，全球智能设备产业就处于高速发展期，移动互联网浪潮席卷而来，智能手机、智能电视等设备快速普及。然而，当时主流的操作系统如安卓和iOS，不仅存在对硬件设备的适配局限，还面临着不同设备间数据互通困难、协同操作体验不佳等问题。同时，国际技术竞争日益激烈，操作系统作为数字经济时代的核心基础设施，其自主可控对企业乃至国家的信息安全和产业发展至关重要。

- 序幕：2018年8月24日，华为申请了“华为鸿蒙”商标，标志着鸿蒙系统开始从技术研发走向商业化布局。到了2019年5月14日，华为鸿蒙商标获得注册公告，专用权限期从2019年5月14日至2029年5月13日，这为鸿蒙系统的商业化推广与应用提供了品牌保障，也正式拉开了鸿蒙系统在智能终端市场上的探索与发展序幕。
- 鸿蒙1.0：2019年8月9日，华为在开发者大会上正式发布鸿蒙1.0系统，这一版本首次应用于华为荣耀智慧屏产品中，标志着华为正式进军操作系统领域。该版本初步展现了分布式能力雏形，为后续系统发展奠定了基础，拉开了鸿蒙系统在智能终端领域探索的序幕。

- 鸿蒙2.0：2020年9月10日发布，其应用范围进一步拓展，适用于部分手机、车机、智能电视等设备。此版本引入全场景设备互联概念，在性能方面也进行了优化，让不同设备之间的协同工作更加顺畅，开始构建起鸿蒙系统的全场景生态框架。
- 鸿蒙3.0：2022年7月27日发布，在超级终端方面有了重大升级，支持更多设备加入，进一步提升了跨设备协同体验。同时，在鸿蒙智联、万能卡片、流畅性能、隐私安全、信息无障碍等方面均有显著提升，全方位提升了用户在多设备使用场景下的体验。
- 鸿蒙4.0：2023年8月4日发布，着重强化了智能互联能力，尤其在多屏跨设备投屏等功能上实现了技术突破，带来更便捷、高效的跨设备交互体验。新增的AI交互功能，也为用户带来了更智能的操作感受，并且支持更多智能设备融入鸿蒙生态。
- 鸿蒙4.4：于2024年推出，作为传统分支的延续版本，它针对耳机等特定设备进行了适配优化，进一步完善了鸿蒙系统在可穿戴设备等细分领域的应用体验，提升了特定设备与其他鸿蒙终端的协同能力。
- 鸿蒙5.0（HarmonyOS NEXT）：2024年10月22日发布，这是具有里程碑意义的版本，它是中国首个实现全栈自研的操作系统，标志着中国在操作系统领域取得了突破性进展。该版本彻底脱离安卓，其流畅度显著提升。2025年3月推送的鸿蒙5.0.3版本，新增“网络邻居”功能，优化了相机和第三方应用的兼容性；鸿蒙5.0.5版本适配Mate 70/X6/Pad等旗舰设备，侧重于性能调优，进行了分层级体验优化。

1.2　HarmonyOS的设计理念

在万物互联的时代，人们每天都会接触多种不同形态的设备。每种设备在特定的场景下能够解决一些特定的问题，表面看起来能够做到的事情更多了，但每种设备在使用时是孤立的，提供的服务也都局限于特定的设备，因此人们的生活并没有变得更好更便捷，反而变得非常复杂。HarmonyOS的诞生旨在解决这些问题，在纷繁复杂的世界中回归本源，建立平衡，连接万物。

混沌初开，一生二，二生三，三生万物，HarmonyOS旨在为用户打造一个和谐的数字世界——One Harmonious Universe。

1. One

万物归一，回归本源。基于以人为本的设计初衷，通过严谨的实验探究背后的人因，并将其结论融入鸿蒙系统的设计当中。

HarmonyOS系统的表现应该符合人的本质需求。为保障全场景多设备的舒适体验，在整个系统中，各种大小的文字都清晰易读，图标精确而清晰、色彩舒适而协调、动效流畅而生动。同时，界面元素层次清晰，能巧妙地突出界面的重要内容，并能传达元素可交互的感觉。另外，系统的表现应该是直接的，用户在使用过程中无须思考。因此，系统的操作需要符合人的本能，并且使用智能化的技术能力主动适应用户的习惯。

2. Harmonious

一生为二，平衡共生。万物皆有两面，虚与实、阴与阳、正与反，等等。二者虽截然不同却可以很好地融合，达到平衡。

HarmonyOS希望给用户带来和谐的视觉体验。通过将光影、材质等设计转化到界面设计中，给用户带来高品质的视觉享受。同时，物理世界中的体验记忆转化到虚拟世界中，熟悉的印象有助于用户快速理解界面元素并完成相应的操作。

3. Universe

三生万物，演化自如。HarmonyOS是面向多设备体验的操作系统，因此，给用户提供舒适便捷的多设备操作体验是 HarmonyOS 区别于其他操作系统的核心要点。

一方面，界面设计、组件设计需要拥有良好的自适应能力，可快速适应不同尺寸屏幕的开发。

另一方面，期望多设备的体验能在一致性与差异性中取得良好的平衡。

- 一致性：界面中的元素设计以及交互方式尽量保持一致，以减少用户的学习成本。
- 差异性：不同类型的设备在屏幕尺寸、交互方式、使用场景、用户人群等方面都会存在一定的差异性，为了给用户提供合适的操作体验，我们需要针对不同类型的设备进行差异化设计。

同时，HarmonyOS作为面向全球用户的操作系统，为了让更多的用户能够享受科技的便利以及具有愉悦的体验，它在数字健康、全球化、无障碍等方面进行了积极的探索与思考。

1.3　HarmonyOS的整体架构

HarmonyOS整体的分层结构自下而上依次为内核层、系统服务层、框架层、应用层，如图1-2所示。HarmonyOS基于多内核设计，系统功能按照“系统→子系统→功能/模块”逐级展开。在多设备部署场景下，各功能模块组织符合“抽屉式”设计，即功能模块采用面向切面编程（Aspect Orient Programming，AOP）的设计思想，可根据实际需求裁剪某些非必要的子系统或功能模块。

图1-2　HarmonyOS整体架构

HarmonyOS实现了模块化耦合，对应不同设备可实现弹性部署，使其可以方便、智能地适配GB、

MB、KB等由高到低的不同内存规模设备，可以便捷地在手机、智慧屏、车机、穿戴设备等IoT设备间实现数据的流转与迁移。

1. 内核层

内核层基于Linux系统设计，主要包括内核子系统和驱动子系统。

- 内核子系统：HarmonyOS采用多内核设计，支持针对不同资源受限设备选用适合的OS内核。KAL（Kernel Abstract Layer，内核抽象层）通过屏蔽多内核差异，为上层提供基础的内核能力，包括进程/线程管理、内存管理、文件系统、网络管理和外设管理等。
- 驱动子系统：硬件驱动框架（HDF）。HarmonyOS驱动框架是HarmonyOS硬件生态开放的基础，提供了统一的外设访问能力和驱动开发、管理框架，如图1-3所示。

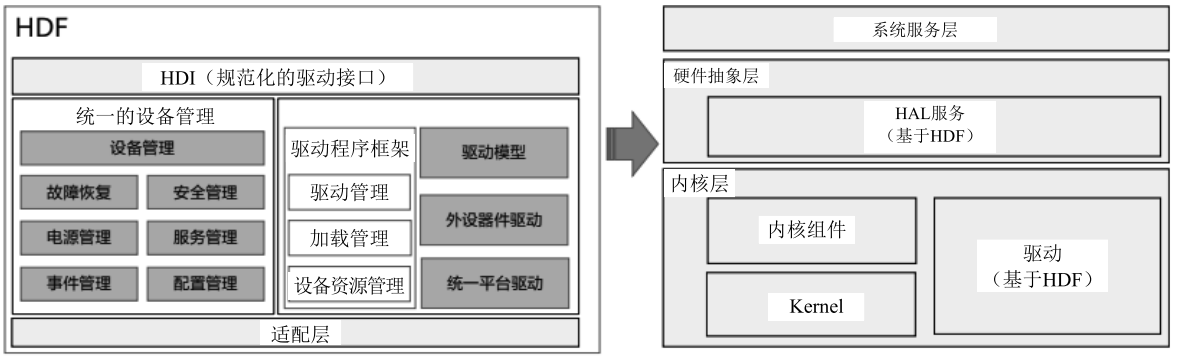

图1-3　内核层驱动子系统

2. 系统服务层

系统服务层是HarmonyOS的核心能力集合，通过框架层对应用程序提供服务。该层架构如图1-4所示，包含以下几个部分：

- 系统基本能力子系统集：为分布式应用在HarmonyOS多设备上的运行、调度、迁移等操作提供了基础能力，由分布式软总线、分布式数据管理、分布式任务调度、方舟多语言运行时、公共基础库、多模输入、图形、安全、AI等子系统组成。其中，方舟运行时提供了C / C++ / JavaScript多语言运行时和基础的系统类库，也为使用方舟编译器静态化的Java程序（即应用程序或框架层中使用Java语言开发的部分）提供运行时。
- 基础软件服务子系统集：为HarmonyOS提供公共的、通用的软件服务，由事件通知、电话、多媒体、DFX、MSDP & DV等子系统组成。
- 增强软件服务子系统集：为HarmonyOS提供针对不同设备的、差异化的能力增强型软件服务，由智慧屏专有业务、穿戴专有业务、IoT专有业务等子系统组成。
- 硬件服务子系统集：为HarmonyOS提供硬件服务，由位置服务、生物特征识别、穿戴专有硬件服务、IoT专有硬件服务等子系统组成。

根据不同设备形态的部署环境，基础软件服务子系统集、增强软件服务子系统集、硬件服务子系统集内部可以按子系统粒度裁剪，每个子系统内部又可以按功能粒度裁剪。

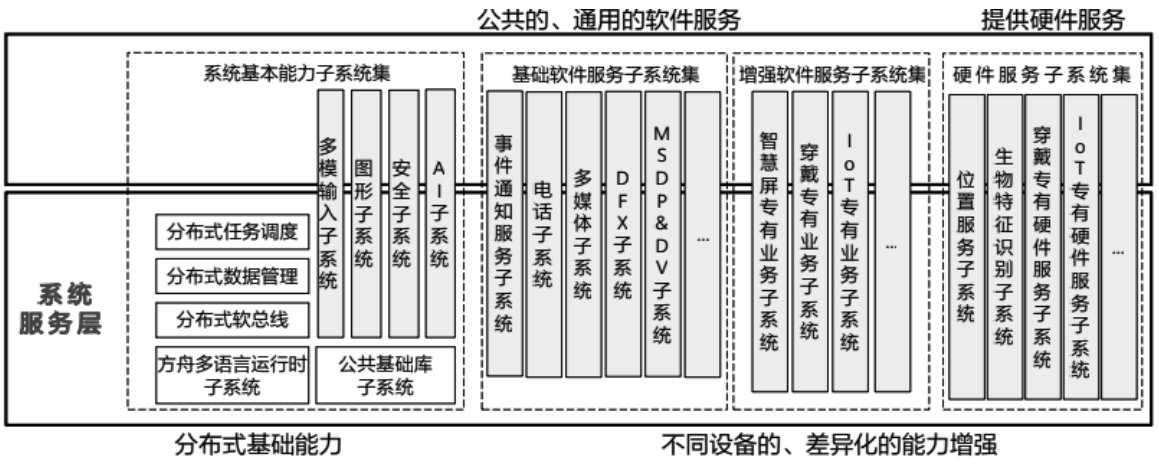

图1-4　HarmonyOS系统服务层架构

3. 框架层

框架层为HarmonyOS的应用程序提供以下支持：

- 用户程序框架：支持Java/C/C++/JavaScript等多种语言。
- Ability框架：应用所具备能力的抽象。
- 两种UI框架：适用于Java语言的Java UI框架和适用于JavaScript语言的JavaScript UI框架。
- 多语言框架API：支持多种软硬件服务对外开放的语言框架。

根据系统的组件化裁剪程度，HarmonyOS设备支持的API也会有所不同。

4. 应用层

应用层包括系统应用和第三方非系统应用，如图1-5所示。

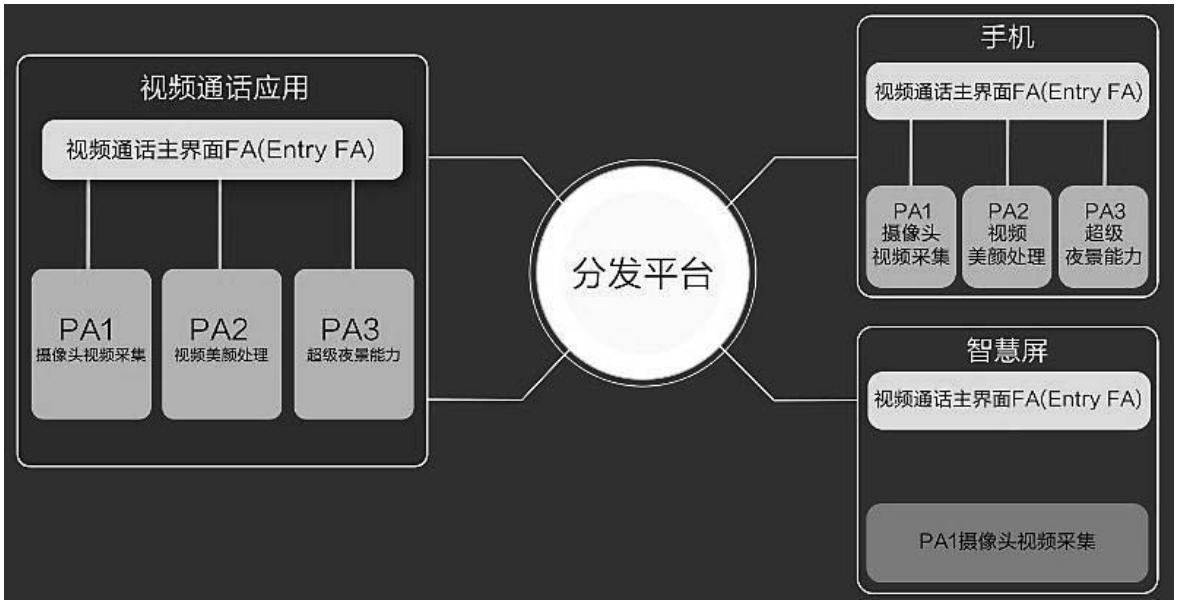

图1-5　HarmonyOS应用层

HarmonyOS的应用由一个或多个FA（Feature Ability）或PA（Particle Ability）组成。

- FA有UI界面，提供与用户交互的能力；而PA无UI界面，提供后台运行任务的能力以及统一的数据访问对象。
- FA在进行用户交互时，所需的后台数据访问也需要由对应的PA提供支撑。
- 基于FA/PA开发的应用，能够实现特定的业务功能，支持跨设备调度与分发，为用户提供一致、高效的应用体验。

1.4　HarmonyOS的技术特性

HarmonyOS具有以下技术特性：

1. 硬件互助，资源共享

1）分布式软总线

分布式软总线是多设备终端的统一基座，为设备间的无缝互联提供了统一的分布式通信能力，能够快速发现并连接设备，高效地传输任务和数据。

2）分布式数据管理

分布式数据管理基于分布式软总线，实现了应用程序数据和用户数据的分布式管理。用户数据不再与单一物理设备绑定，业务逻辑与数据存储分离，应用跨设备运行时数据无缝衔接，为打造一致、流畅的用户体验创造了基础条件。

3）分布式任务调度

分布式任务调度基于分布式软总线、分布式数据管理、分布式Profile等技术特性，构建统一的分布式服务管理（发现、同步、注册、调用）机制，支持对跨设备的应用进行远程启动、远程调用、绑定/解绑、迁移等操作，能够根据不同设备的能力、位置、业务运行状态、资源使用情况并结合用户的习惯和意图，选择最合适的设备运行分布式任务。

4）设备虚拟化

分布式设备虚拟化平台可以实现不同设备的资源融合、设备管理、数据处理，将周边设备作为手机能力的延伸，共同形成一个超级虚拟终端。

2. 一次开发，多端部署

HarmonyOS提供用户程序框架、Ability框架以及UI框架，能够保证开发的应用在多终端运行时保持一致性。一次开发；多端部署。

多终端软件平台API具备一致性，确保用户程序的运行兼容性。

- 支持在开发过程中预览终端的能力适配情况（CPU、内存、外设、软件资源等）。
- 支持根据用户程序与软件平台的兼容性来调度用户程序。

3. 统一OS，弹性部署

HarmonyOS通过组件化和组件弹性化等设计方法，做到硬件资源的可大可小，在多种终端设备间，按需弹性部署，全面覆盖了ARM、RISC-V、x86等各种CPU，以及从KB到GB级别的RAM。

1.5　HarmonyOS的应用场景

HarmonyOS以手机为核心，构建1+8+N全场景应用，如图1-6所示。

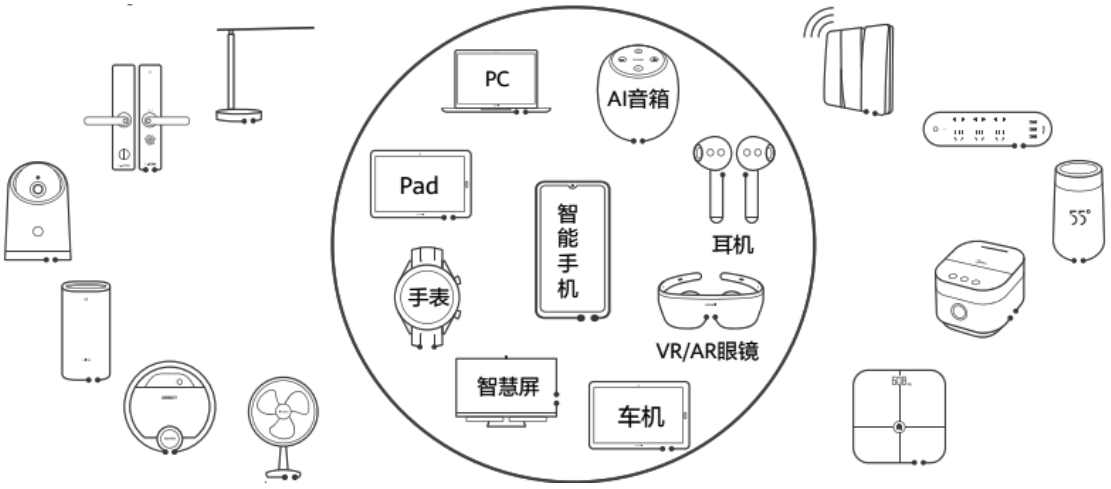

图1-6　HarmonyOS 1+8+N全场景应用

HarmonyOS典型的应用场景如表1-1所示。

表1-1　HarmonyOS典型的应用场景

设备类型	具体实现	技术支撑
智能家居	实现灯光、温控等设备的跨屏协同控制，通过手机/平板统一管理全屋生态	分布式软总线技术
车载系统	车机与手机无缝流转导航/音乐，仪表盘可实时显示手机日程或健康数据	原子化服务跨端调用
工业设备	支持工业机器人、自动化产线的远程监控与协同调度，提供标准化控制接口	低时延通信协议
穿戴设备	与手机协同实现健康监测报警，数据自动同步至云端健康档案	轻量化服务卡片

第 2 章

应用开发准备

上一章简要介绍了HarmonyOS系统，本章将继续介绍鸿蒙应用开发的准备工作，包括下载和安装DevEco Studio开发工具、DevEco Studio的用法、应用工程结构以及应用/服务开发流程。华为鸿蒙DevEco Studio 是面向全场景的一站式集成开发环境（Integrated Development Environment，IDE），支持全场景多设备，提供集分布式多端开发、分布式多端调测、多端模拟仿真于一体的研发平台，并提供全方位的质量与安全保障。

2.1 开发环境搭建

俗话说：“工欲善其事，必先利其器。”为了顺利进行HarmonyOS应用开发，需要事先准备好DevEco Studio开发工具，即HarmonyOS的一站式集成开发环境。

2.1.1 下载DevEco Studio

下面以在Windows系统中安装 DevEco Studio为例，介绍如何下载、安装和配置开发环境。
为保证DevEco Studio正常运行，建议计算机配置满足如下要求：

- 操作系统：Windows 10 64位。
- 内存：8GB及以上。
- 硬盘：100GB及以上。
- 分辨率：1280 × 800像素及以上。

具体操作步骤如下：

01 打开 DevEco Studio官方网站，在页面上单击“立即下载”按钮，如图2-1所示。
02 单击“立即下载”按钮，会跳转到华为账号登录界面，如图2-2所示。
03 注册账号后继续下载，DevEco Studio提供了Windows版本和Mac版本，可以根据操作系统选择对应的版本进行下载。笔者下载的是DevEco Studio 5.0.5 Release版本，版本信息如图2-3所示。

图2-1　HarmonyOS官方网页

图2-2　华为账号登录页面

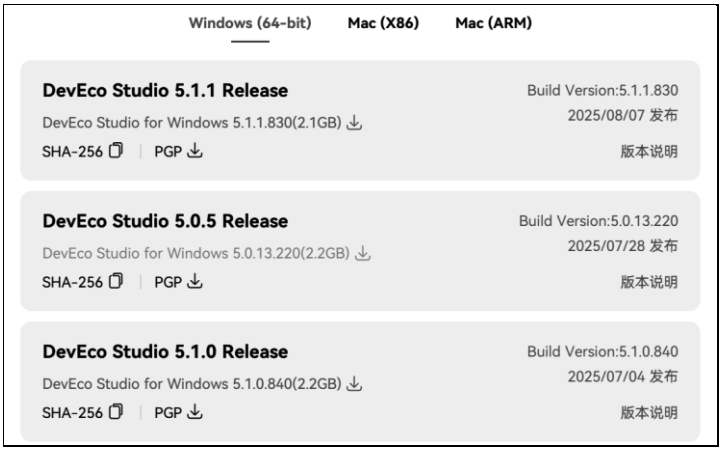

图2-3　DevEco Studio下载版本

2.1.2　安装DevEco Studio

01 解压下载的DevEco Studio压缩包，启动exe安装程序，进入安装欢迎界面，如图2-4所示。

02 单击“下一步”按钮，进入“选择安装位置”界面，选择安装路径，这里根据自己的喜好选择即可，需要约10GB的存储空间，如图2-5所示。

03 单击“下一步”按钮，进入“安装选项”界面，勾选“DevEco Studio”和“添加‘bin’文件夹到PATH”复选框，如图2-6所示。

图 2-4　DevEco Studio 安装欢迎界面

图 2-5　DevEco Studio 安装路径

04 单击“下一步”按钮，进入“选择开始菜单目录”，选择开始菜单文件夹，这里保持默认即可，如图2-7所示。

图 2-6　DevEco Studio 安装选项

图 2-7　DevEco Studio 选择开始菜单目录

05 单击“下一步”按钮，进入安装程序结束界面，勾选“运行DevEco Studio”复选框，并单击“完成”按钮，如图2-8所示。

图2-8　DevEco Studio安装完成

至此，DevEco Studio安装完成。

2.2　创 建 工 程

使用DevEco Studio创建工程（Project，本书翻译为工程）的步骤如下：

01 创建工程。若是首次打开DevEco Studio，则在欢迎页面单击Create Project创建工程，如图2-9所示。如果已经打开了一个工程，则在菜单栏中依次单击File→New→Create Project来创建一个新工程。

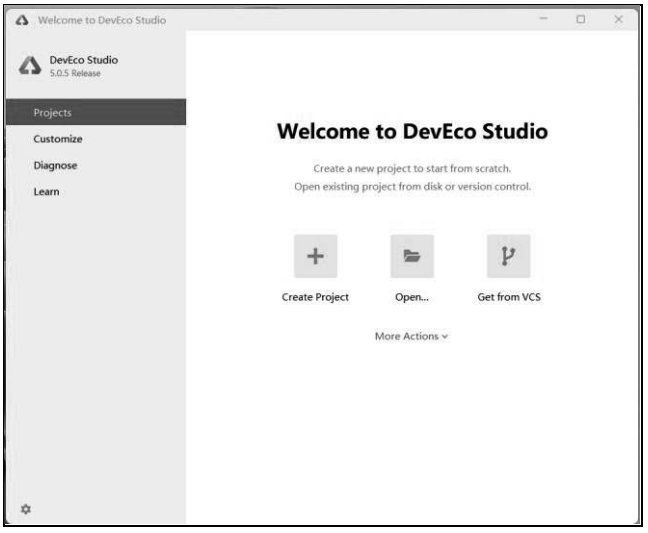

图2-9　DevEco Studio首页

02 选择Application项目。进入选择能力模板页面，在页面左侧选择Application，然后在页面右侧选择Empty Ability，再单击Next按钮，如图2-10所示。

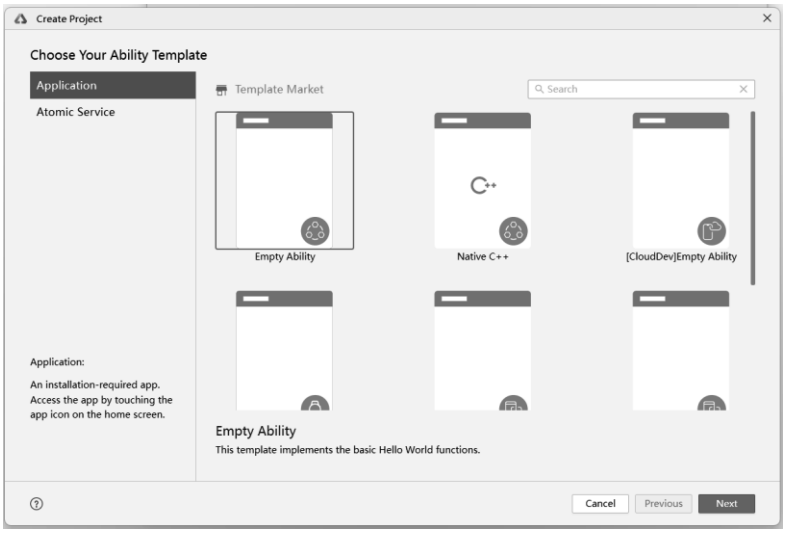

图2-10　DevEco Studio创建项目

03 配置工程。进入工程配置页，如图2-11所示。

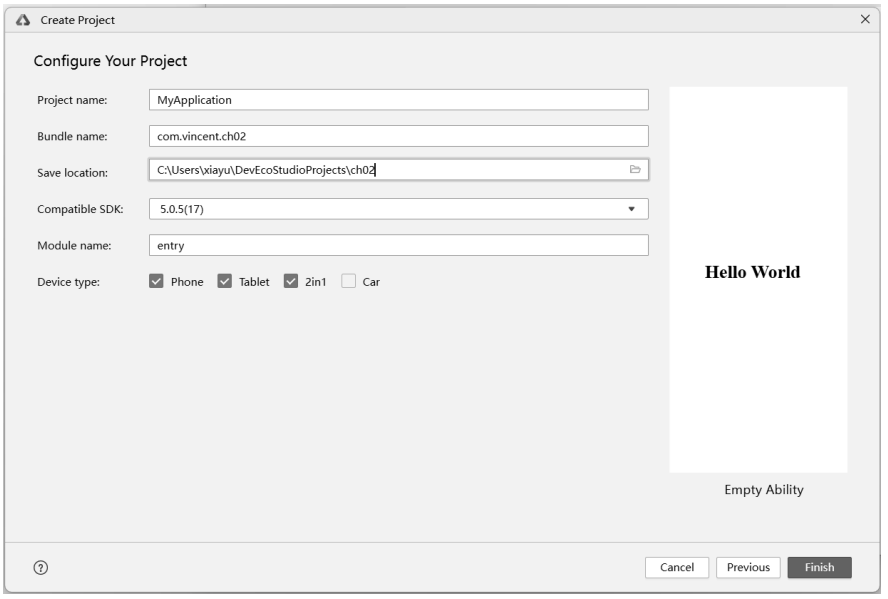

图2-11　工程配置信息

在工程配置页中，几个关键配置项的详细信息说明如下：

- Project name：开发者可以自行设置的项目名称。
- Bundle name：包名称，默认情况下应用ID也会使用该名称，应用发布时对应的ID需要保持一致。
- Save location：工程保存路径，建议用户自行设置相应位置。
- Compile SDK：编译的API版本，这里默认选择5.0.5。
- Model name：模型名称，保持默认即可。

04 成功创建工程框架。配置完成后，单击Finish按钮，DevEco Studio会自动生成示例代码和相关资源，完成工程框架的创建。

2.3　DevEco Studio界面简介

进入DevEco Studio（IDE）后，我们首先了解一下它的基础界面。整个界面大致上可以分为4个部分，分别是工程目录区、代码编辑区、预览区以及通知栏，如图2-12所示。

1. 工程目录区

界面左侧为工程目录区，后续章节将会详细介绍。

2. 代码编辑区

界面中间的是代码编辑区，可以在这里修改代码以及切换显示的文件。通过按住Ctrl键并滚动鼠标滚轮，可以实现界面的放大与缩小。

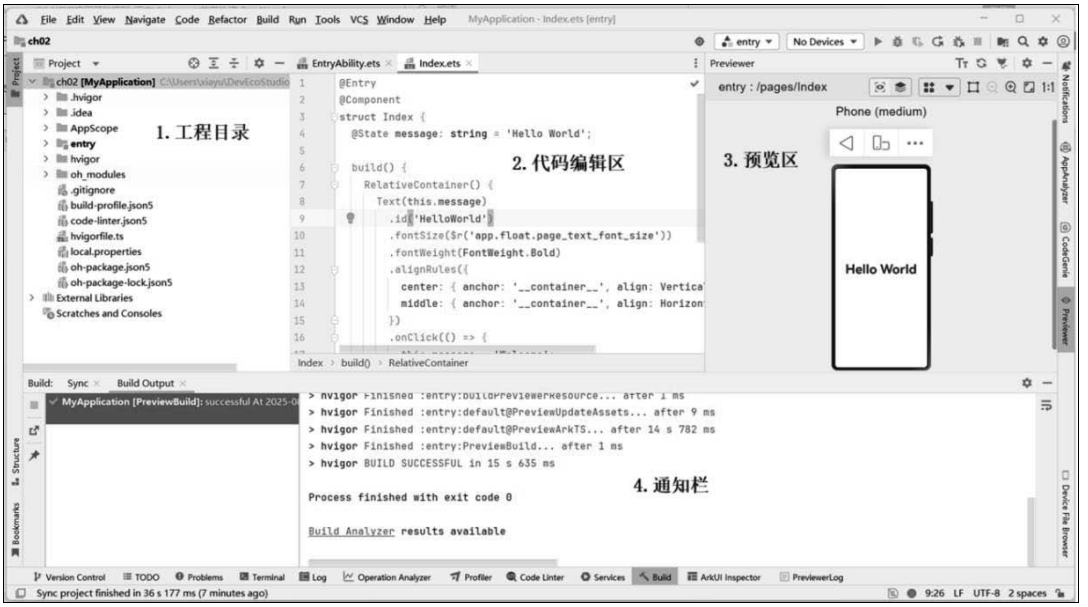

图2-12　DevEco Studio界面介绍

3. 预览区

单击DevEco Studio右侧的Previewer按钮，可以预览相应文件的UI展示效果，如图2-13所示。

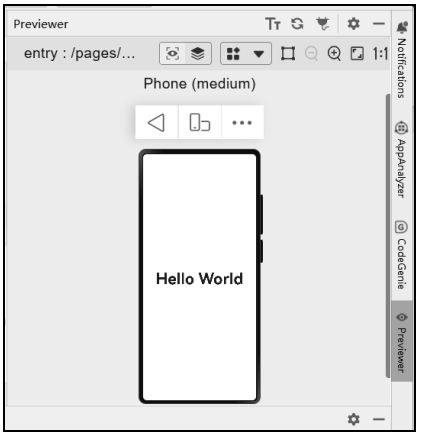

图2-13　竖屏预览

预览器提供了一些基本功能，包括旋转屏幕、切换显示设备及多设备预览等。单击旋转按钮，可以切换竖屏和横屏的显示效果，如图2-14所示。也可以单击图2-14中框选的按钮，切换显示的设备类型。弹出框内会显示Available Profiles，即可用的设备类型，如图2-15所示。如单击Foldable切换设备，也可以单击旋转按钮切换Foldable的横竖屏显示模式。

打开 Muti-profile preview 开关，可以实现多个尺寸设备的实时预览，如图2-16所示。

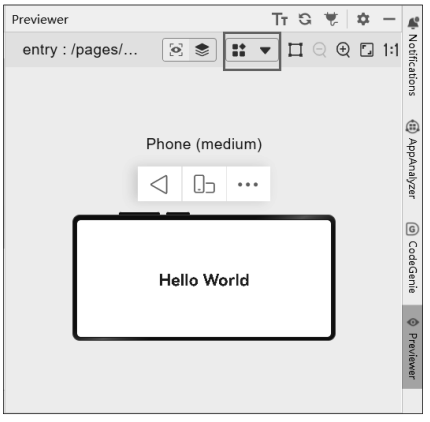

图 2-14　横屏预览

图 2-15　切换设备

单击预览器右上角的组件预览按钮，可以进入组件预览界面，如图2-17所示。

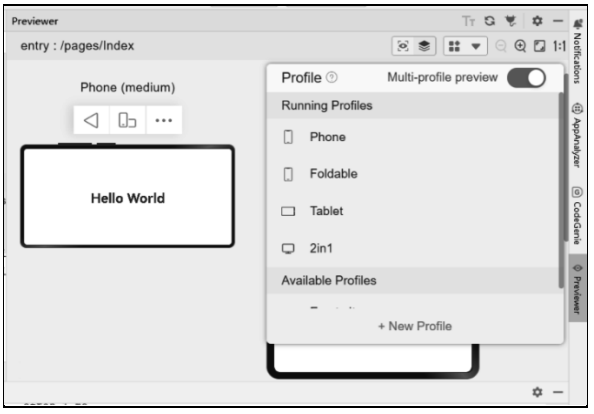

图 2-16　多尺寸设备预览

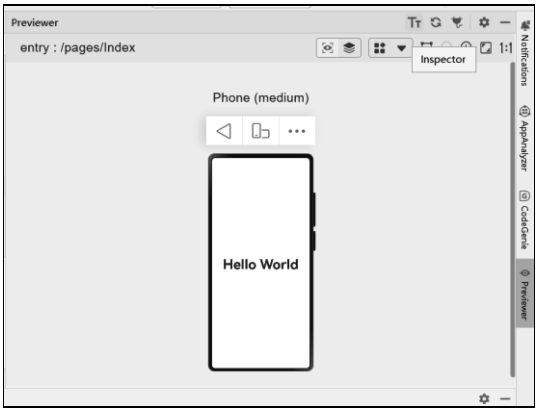

图 2-17　组件预览模式

在组件预览模式下可以预览当前组件对应的代码块。在预览界面中单击相应组件，则在代码文件中会框选对应的组件代码，右侧组件树则展示当前组件的基本属性，如图2-18所示。

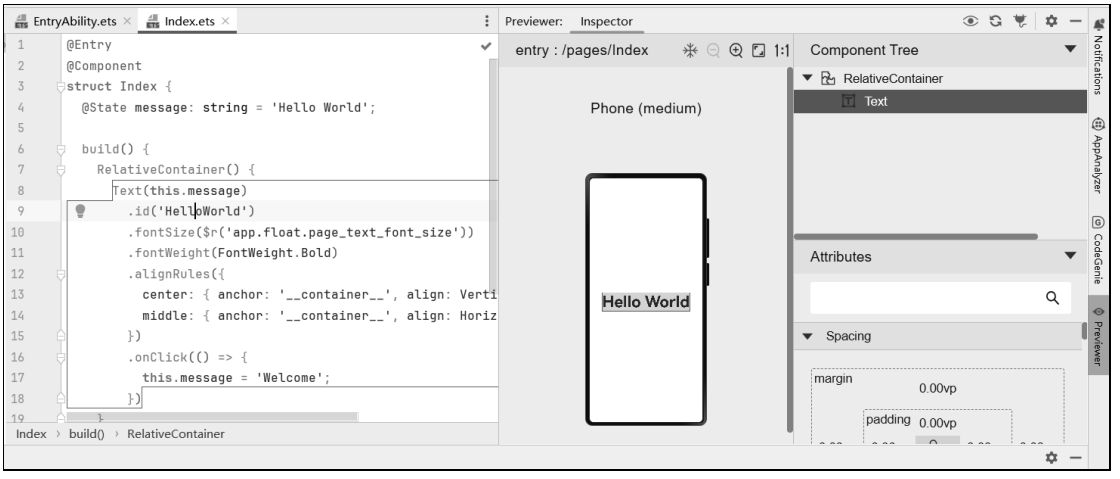

图2-18　组件属性预览

4. 通知栏

在编辑器底部有一行工具，单击这些工具，对应的信息将在通知栏中显示。其中常用的有：Run用于显示项目运行时的信息；Build显示程序编译情况；Problems用于显示当前工程错误与提醒信息；Terminal是命令行终端，在这里执行命令行操作；PreviewerLog用于输出预览器日志输出；Log用于输出模拟器和真机运行时的日志。在后续使用中会陆续接触这些工具。

2.4　运行Hello World工程

DevEco Studio提供本地模拟器供开发者（读者，下文同）模拟运行本地开发的工程，本节将以Hello World为例，演示一下工程运行方法。首先需要在DevEco Studio中安装本地模拟器，然后运行。

01 依次单击顶部工具栏菜单中的Tools→Device Manager，如图2-19所示。

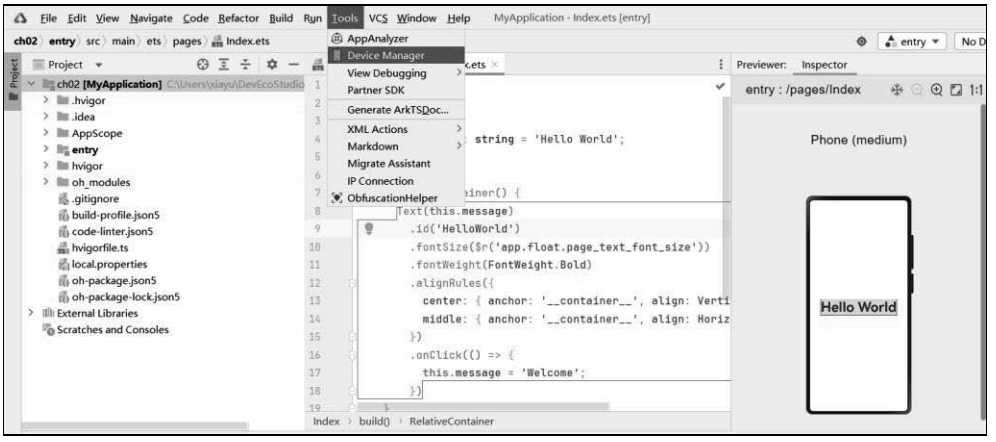

图2-19　IDE工具栏

02 选择Local Emulator，安装模拟器（需要登录），如图2-20所示。

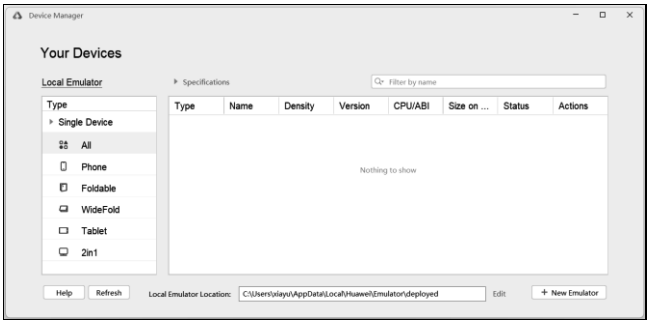

图2-20　模拟器列表

03 设置合适的 Local Emulator Location 存储地址，然后单击左下角的“+ New Emulator”按钮，进入选择模拟器界面，如图2-21所示。

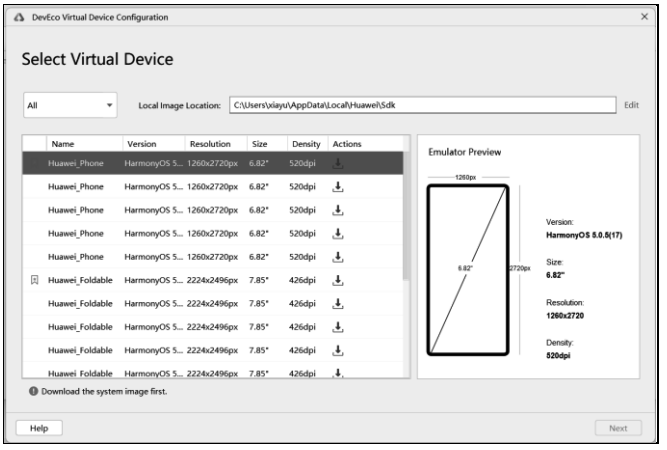

图2-21　模拟器列表

04 选择Huawei_Phone手机模拟器，单击下载箭头，进入下载模拟器界面，如图2-22所示。

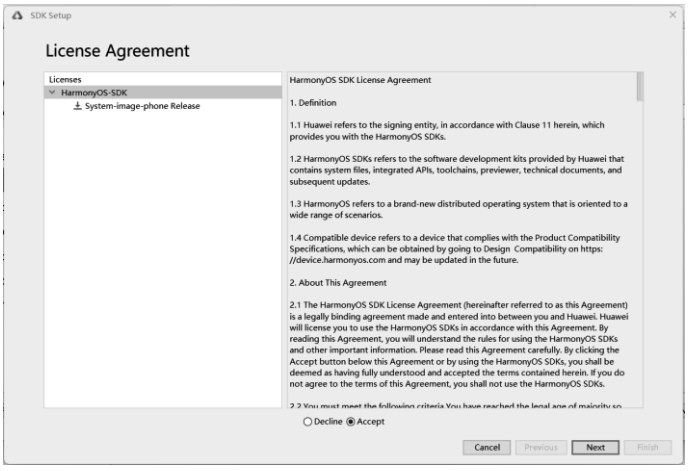

图2-22　下载模拟器界面

05 单击Accept单选按钮，再单击Next按钮，等待下载完成，如图2-23所示。

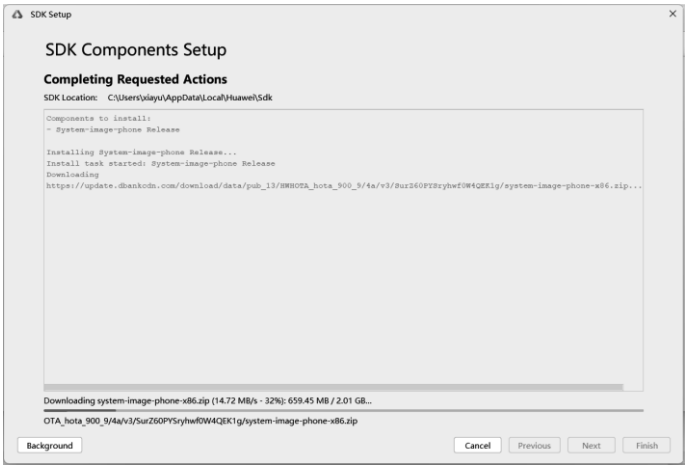

图2-23　模拟器下载界面

06 下载完成后，即可创建相应的手机模拟器，单击Finish按钮完成创建，如图2-24所示。

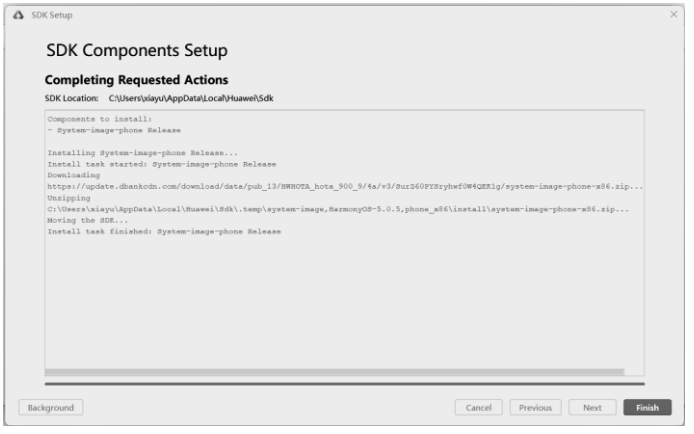

图2-24　模拟器下载完成界面

07 单击Next按钮，进入手机模拟器配置界面，如图2-25所示。

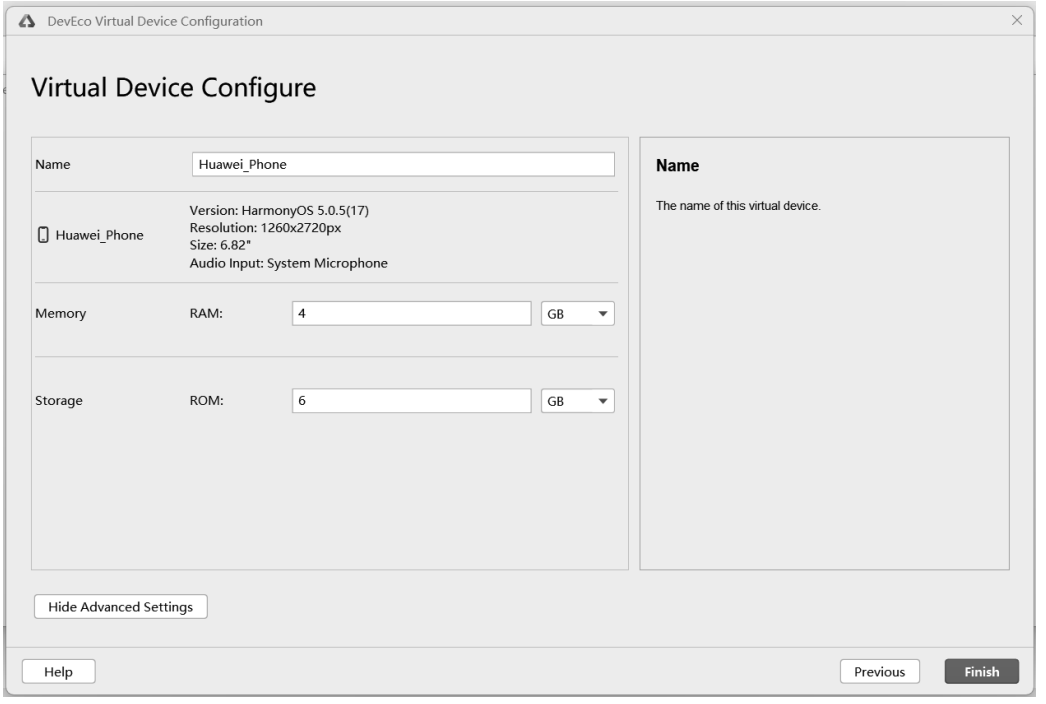

图2-25　模拟器配置界面

08 保持默认配置，单击Finish按钮，完成配置，此时的模拟器管理界面如图2-26所示。

09 配置完成后，在Local Emulator页面中会出现创建的手机模拟器，单击左边Actions列下面的小三角图标按钮，就能够启动模拟器，如图2-27所示。

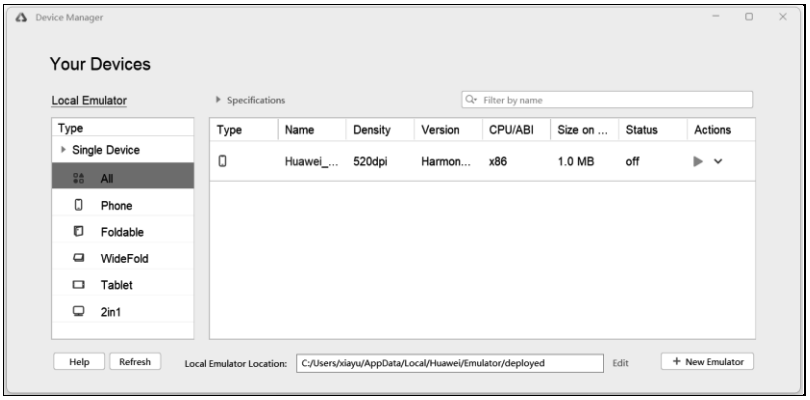

图 2-26　模拟器管理界面

图 2-27　模拟器启动后的界面

10 模拟器启动后，在IDE右上角单击绿色的小三角启动按钮，就能将Hello World工程运行到模拟器上，如图2-28所示。

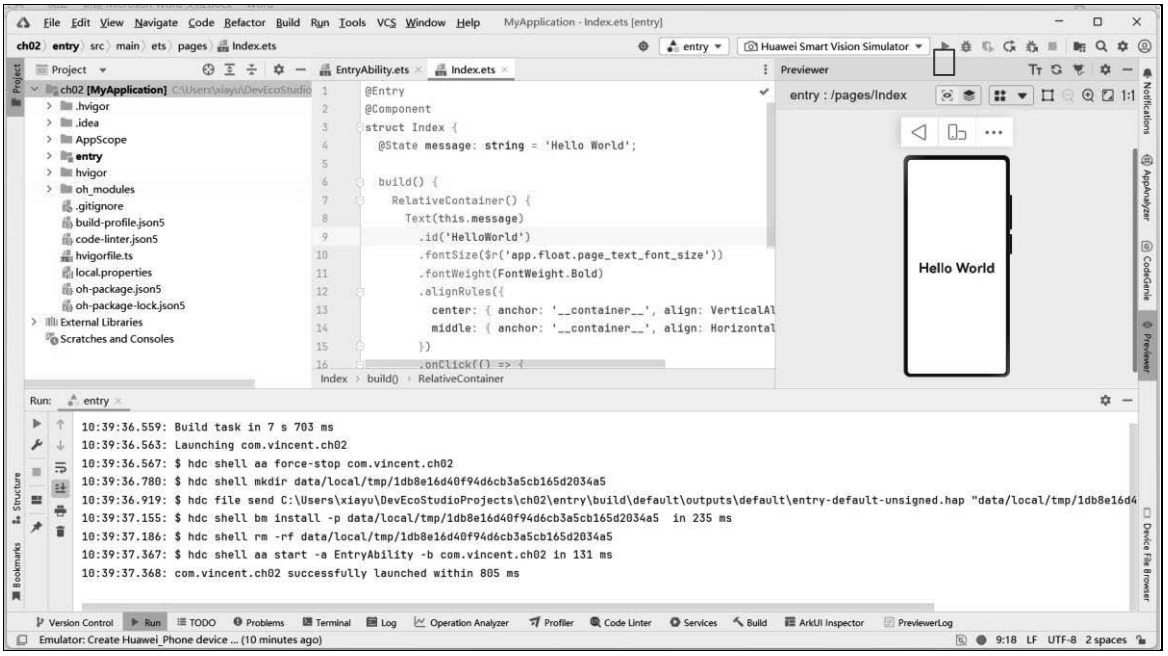

图2-28　Hello World运行界面

运行效果如图2-29所示。

图2-29　Hello World运行结果

2.5　应用工程结构介绍

本节以2.4节的Hello World工程为例，介绍应用工程结构。

2.5.1　工程级目录

2.4节的Hello World工程的目录结构如图2-30所示。其中几个主要目录介绍如下：

- AppScope：用于存放应用全局所需的资源文件，其中有resources文件夹和配置文件app.json5（在2.5.3节详细介绍）。AppScope>resources>base中又包含element和media两个文件夹，如图2-31所示。
  - Element：用于存放公共的字符串、布局文件等资源。
  - media：用于存放全局公共的多媒体资源文件。

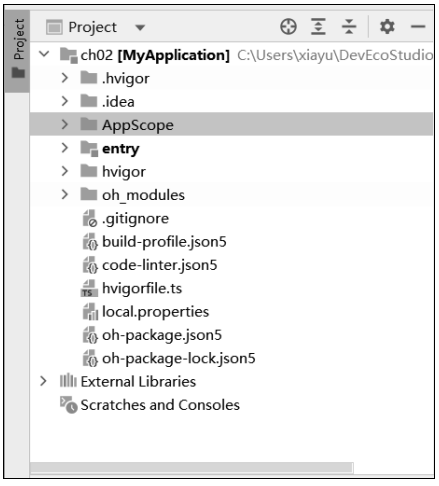

图 2-30　工程目录结构图

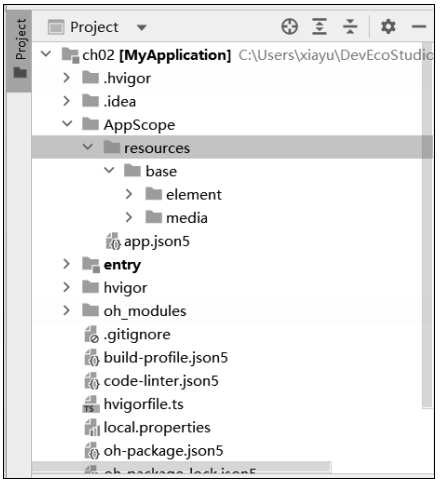

图 2-31　多媒体资源目录

- entry：应用的主模块，用于存放HarmonyOS应用的代码、资源等，并编译构建生成一个HAP包。
- oh-package.json5：用来描述包名、版本、入口文件（类型声明文件）和依赖项等信息。
- oh_modules：工程的依赖包，用于存放三方库依赖信息。
- build-profile.json5：工程级配置信息，包括签名、产品配置等。
- hvigorfile.ts：工程级编译构建任务脚本。hvigor是基于任务管理机制实现的一款全新的自动化构建工具，主要提供任务注册编排、工程模型管理、配置管理等核心能力。
- oh-package.json5：工程级依赖配置文件，用于记录引入包的配置信息。

2.5.2　模块级目录

模块级目录entry如图2-32所示，其中几个主要目录介绍如下：

- src>main>ets：用于存放ets代码。
- src>main>resources：用于存放模块内的多媒体及布局文件等。
- src>main>module.json5：为模块的配置文件，在2.5.4节详细介绍。
- ohosTes：是单元测试目录。
- build-profile.json5：是模块级配置信息，包括编译和构建配置项。
- hvigorfile.ts：是模块级构建脚本。
- oh-package.json5：是模块级依赖配置信息文件。

进入src > main > ets目录，其中有entryability、entrybackupabilit、pages三个文件夹。

- entryability：用于存放ability文件，负责当前ability应用逻辑和生命周期管理。

- entrybackupability：为应用提供扩展的备份恢复能力。
- pages：用于存放UI界面相关代码文件，初始会生成一个Index页面，如图2-33所示。

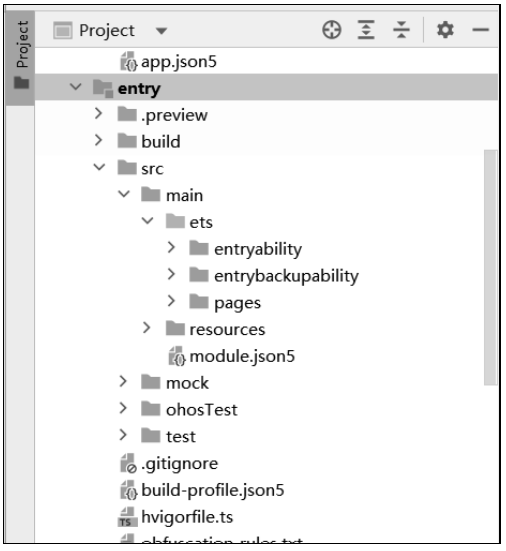

图 2-32　entry 目录结构

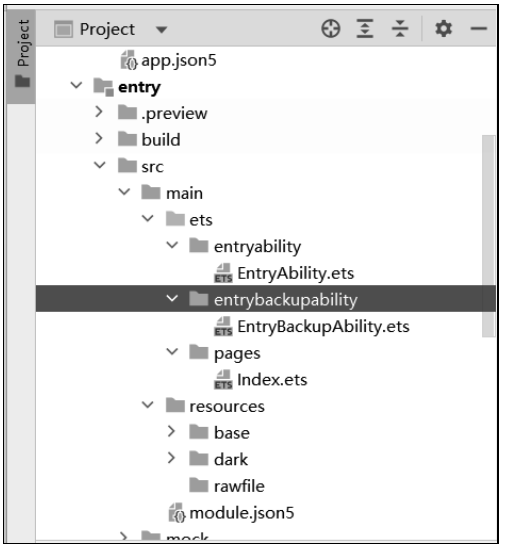

图 2-33　page 页面目录结构

2.5.3　app.json5

AppScope > app.json5是应用的全局配置文件，用于存放应用公共的配置信息，如图2-34所示。

图2-34　全局配置文件app.json5

其中主要的配置信息如下：

- bundleName：包名。
- vendor：应用程序供应商。
- versionCode：用于区分应用版本。
- versionName：版本号。

2.5.4　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，如图2-35所示。

图2-35　模块配置文件module.json5

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。其具体属性及描述可以参照表2-1。

表2-1　module.json5配置文件标签说明

属性名称	含　义	数据类型	是否可缺省
name	标识当前模块的名称，确保该名称在整个应用中唯一。取值为长度不超过128字节的字符串（当前DevEco新建Module时，name最大长度为31字节。后续DevEco版本支持name长度最大128字节），不支持中文。 应用升级时允许修改该名称，但需要应用适配Module相关数据目录的迁移，详见文件管理接口	字符串	该标签不可缺省
type	标识当前模块的类型。支持的取值如下： - entry：应用的主模块。 - feature：应用的动态特性模块。 - har：静态共享包模块。 - shared：动态共享包模块	字符串	该标签不可缺省
srcEntry	标识当前模块所对应的代码路径，取值为长度不超过127字节的字符串	字符串	该标签可缺省，默认值为空
description	标识当前模块的描述信息，取值为长度不超过255字节的字符串，可以采用字符串资源索引格式	字符串	该标签可缺省，默认值为空
process	标识当前模块的进程名，取值为长度不超过31字节的字符串。如果在HAP标签下配置了process，则该应用的所有UIAbility、DataShareExtensionAbility、ServiceExtensionAbility都运行在该进程中。 说明：仅支持系统应用配置，三方应用配置不生效	字符串	该标签可缺省，默认为app.json5文件下app标签下的bundleName
mainElement	标识当前模块的入口UIAbility名称或者ExtensionAbility名称，取值为长度不超过255字节的字符串	字符串	该标签可缺省，默认值为空

（续表）

属性名称	含　义	数据类型	是否可缺省
deviceTypes	标识当前模块可以运行在哪类设备上	字符串数组	该标签不可缺省
deliveryWithInstall	标识当前模块是否在用户主动安装的时候安装，即该模块对应的HAP是否跟随应用一起安装。 - true：主动安装时安装。 - false：主动安装时不安装	布尔值	该标签不可缺省
installationFree	标识当前模块是否支持免安装特性。 - true：表示支持免安装特性，且符合免安装约束条件。 - false：表示不支持免安装特性。 说明：当bundleType为元服务时，该字段需要配置为true；反之，该字段需要配置为false	布尔值	该标签不可缺省
virtualMachine	标识当前模块运行的目标虚拟机类型，供云端分发使用，如应用市场和分发中心。如果目标虚拟机类型为ArkTS引擎，则其值为“ark+版本号”	字符串	该标签由IDE构建HAP的时候自动插入
pages	标识当前模块的profile资源，用于列举每个页面信息，取值为长度不超过255字节的字符串	字符串	在有UIAbility的场景下，该标签不可缺省
metadata	标识当前模块的自定义元信息，可通过资源引用的方式配置distributionFilter、shortcuts等信息。只对当前Module、UIAbility、ExtensionAbility生效	对象数组	该标签可缺省，默认值为空
abilities	标识当前模块中UIAbility的配置信息，只对当前UIAbility生效	对象数组	该标签可缺省，默认值为空
extensionAbilities	标识当前模块中ExtensionAbility的配置信息，只对当前ExtensionAbility生效	对象数组	该标签可缺省，默认值为空
definePermissions	标识系统资源HAP定义的权限，不支持应用自定义权限	对象数组	该标签可缺省，默认值为空
requestPermissions	标识当前应用运行时需向系统申请的权限集合	对象数组	该标签可缺省，默认值为空
testRunner	标识用于测试当前模块的测试框架的配置	对象	该标签可缺省，默认值为空
atomicService	当前应用是元服务时，标识相关元服务的配置	对象	该标签可缺省，默认值为空
dependencies	标识当前模块运行时依赖的共享库列表	对象数组	该标签可缺省，默认值为空
targetModuleName	标识当前包所指定的目标模块，确保该名称在整个应用中唯一。取值为长度不超过31字节的字符串，不支持中文。配置该字段的Module具有overlay特性。仅在动态共享包（HSP）中适用	字符串	该标签可缺省，默认值为空
targetPriority	标识当前模块的优先级，取值范围为1~100。配置targetModuleName字段之后，才需要配置该字段。仅在动态共享包（HSP）中适用	整型数值	该标签可缺省，默认值为1
proxyData	标识当前模块提供的数据代理列表	对象数组	该标签可缺省，默认值为空

（续表）

属性名称	含　义	数据类型	是否可缺省
isolationMode	标识当前模块的多进程配置项。支持的取值如下： - nonisolationFirst：优先在非独立进程中运行。 - isolationFirst：优先在独立进程中运行。 - isolationOnly：只在独立进程中运行。 - nonisolationOnly：只在非独立进程中运行	字符串	该标签可缺省，默认值为nonisolationFirst
generateBuildHash	标识当前HAP/HSP是否由打包工具生成哈希值。当配置为true时，在系统OTA升级时应用versionCode保持不变，可根据哈希值判断应用是否需要升级。 该字段仅在app.json5文件中的generateBuildHash字段为false时使能。说明：该字段仅对预置应用生效	布尔值	该标签可缺省，默认值为false
compressNativeLibs	标识libs库是否以压缩存储的方式打包到HAP。 - true：libs库以压缩方式存储。 - false：libs库以不压缩方式存储	布尔值	该标签可缺省，默认值为false
libIsolation	用于区分同应用不同HAP下的.so文件，以防止.so冲突。 - true：当前HAP的.so文件会储存在libs目录中以模块名命名的路径下。 - false：当前HAP的.so文件会直接储存在libs目录中	布尔值	该标签可缺省，默认值为false
fileContextMenu	标识当前HAP的快捷菜单配置项。取值为长度不超过255字节的字符串	字符串	该标签可缺省，默认值为空
querySchemes	标识允许当前应用进行跳转查询的URL schemes，只允许entry类型模块配置，最多50个，每个字符串取值不超过128字节	字符串数组	该标签可缺省，默认值为空
routerMap	标识当前模块配置的路由表路径。取值为长度不超过255字节的字符串	字符串	该标签可缺省，默认值为空
appEnvironments	标识当前模块配置的应用环境变量，只允许entry和feature模块配置	对象数组	该标签可缺省，默认值为空
appStartup	标识当前模块启动框架配置路径，仅在entry中生效	字符串	该标签可缺省，默认值为空
hnpPackages	标识当前应用包含的Native软件包信息。只允许entry类型模块配置	对象数组	该标签可缺省，默认值为空

具体标签内容详见官方网站中有关module.json5配置文件的说明文档。

2.5.5　main_pages.json

src/main/resources/base/profile/main_pages.json文件保存的是页面的路径配置信息，所有需要进行路由跳转的页面都要在这里配置，如图2-36所示。

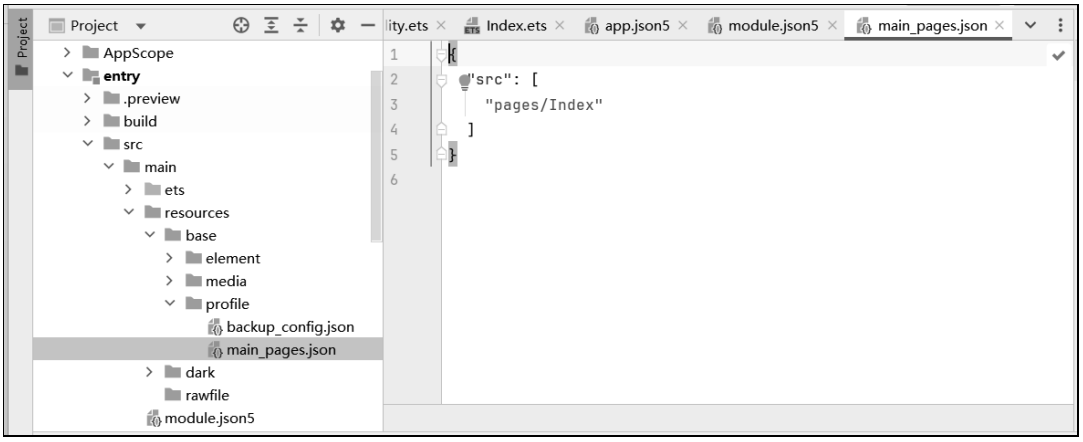

图2-36　界面展示配置文件main_pages.json

2.6　应用/服务开发流程

使用DevEco Studio，只需要按照如下几个步骤，即可轻松开发并上架一个应用/服务（APP/Service）到华为应用市场，如图2-37所示。

图2-37　开发并上架应用流程

（1）开发准备。获取HUAWEI DevEco Studio，完成开发工具的安装及环境配置。

（2）开发应用/服务。DevEco Studio集成了Phone、Tablet、2in1、Car等设备的典型场景模板，可以通过工程向导轻松地创建一个新的工程。

接下来还需要定义应用/服务的UI、开发业务功能等编码工作，可以根据HarmonyOS应用开发概述来查看具体的开发过程，通过API接口文档查阅需要调用的API接口。

在开发代码的过程中，可以使用预览器查看应用/服务效果，支持实时预览、动态预览、双向预览等功能，使编码的过程更高效。

（3）运行、调试和测试应用/服务。应用/服务开发完成后，可以使用真机进行调试（需要申请调测证书进行签名），支持单步调试、跨语言调试等调试手段，使得应用/服务调试更加高效。

HarmonyOS应用/服务开发完成后，在发布到应用/服务市场前，还需要对应用进行测试，主要包含Instrument Test、Local Test，以确保HarmonyOS应用/服务纯净、安全，给用户带来更好的使用体验。

（4）发布应用/服务。HarmonyOS应用/服务开发、测试完成后，需要将应用/服务发布至应用市场，以便应用市场对应用/服务进行分发，普通消费者可以通过应用市场获取到对应的HarmonyOS应用/服务。需要注意的是，发布到华为应用市场的HarmonyOS应用/服务，必须使用应用市场颁发的发布证书进行签名。