

编译 Android

请按照以下说明编译 Android。

设置环境

使用 `envsetup.sh` 脚本初始化环境：

```
source build/envsetup.sh
```

或

```
. build/envsetup.sh
```

有关相关命令的说明，请参阅 [platform/build/envsetup.sh](https://android.googlesource.com/platform/build/+refs/heads/master/envsetup.sh) (<https://android.googlesource.com/platform/build/+refs/heads/master/envsetup.sh>) 中的脚本，包括用于选择设备定位条件的 `lunch`，以及用于编译未捆绑应用的 `tapas`，例如参考 [TV 应用](#) ([/devices/tv/reference-tv-app](#))。

您需要在每次运行 `repo sync` 后重新发出此命令，以获取对该脚本所做的任何更改。请注意，将 `source` 替换为 `.`（一个点）可以省去一些字符，这种简写形式在文档中更为常用。

`envsetup.sh` 脚本会导入若干命令，让您能够使用 Android 源代码，其中包括本练习中使用的命令。以下是一些重要的命令示例：

- **lunch** - `lunch product_name-build_variant` 会选择 ***product_name*** 作为要编译的产品，并选择 ***build_variant*** 作为要编译的变体，然后将这些选择存储在环境中，以便被后续对 `m` 和其他类似命令的调用读取。
- **m** - 从树的顶部运行编译版本。这很有用，因为您可以在子目录中运行 `make`。如果您设置了 `TOP` 环境变量，它便会使用此变量。如果您未设置此变量，它便会从当前目录中查找相应的树，以尝试找到树的顶层。您可以通过运行不包含参数的 `m` 来编译整个源代码树，也可以通过指定相应名称来编译特定目标。
- **mma** - 编译当前目录中的所有模块及其依赖项。

- **mma** - 编译提供的目录中的所有模块及其依赖项。
- **croot** - cd 到树顶部。

要查看可用命令的完整列表，请运行以下命令：

```
hmm
```

选择目标

使用 **lunch** 选择要编译的目标。确切的配置可作为参数进行传递。例如，以下命令表示针对模拟器进行完整编译，并且启用所有调试功能：

```
lunch aosp_arm-eng
```

如果在没有参数的情况下运行，则 **lunch** 会提示您从菜单中选择目标。要了解所有现有设备的编译配置，请参阅[选择设备编译系统 \(/setup/build/running#selecting-device-build\)](#)一文。

所有编译目标都采用 **BUILD-BUILDTYPE** 形式，其中 **BUILD** 是表示特定功能组合的代码。 **BUILDTYPE** 是以下类型之一。

编译类型 使用情况

user	权限受限；适用于生产环境
------	--------------

userdebug	与“user”类似，但具有 root 权限和调试功能；是进行调试时的首选编译类型
-----------	--

eng	具有额外调试工具的开发配置
-----	---------------

userdebug 版本应和 user 版本以相同的方式运行，且能够启用通常不符合平台安全模型的额外调试。这就使得 userdebug 版本具有更强大的诊断功能，因此是进行 user 测试的最佳选择。使用 userdebug 编译系统进行开发时，请遵循 [userdebug 指南 \(/setup/develop/new-device#userdebug-guidelines\)](#)。

eng 版本会优先考虑在平台上工作的工程师的工程生产力。eng 版本会关闭用于提供良好用户体验的各种优化。除此之外，eng 版本的运行方式类似于 user 和 userdebug 版本，以便设备开发者能够看到代码在这些环境下的运行方式。

要详细了解如何针对实际硬件进行编译以及如何在实际硬件上运行编译版本，请参阅[运行版本 \(/setup/build/running\)](#)。

编译代码

本部分是一个简短摘要，用于确保设置已完成。

使用 `m` 编译所有内容。`m` 可以使用 `-jN` 参数处理并行任务。如果您没有提供 `-j` 参数，编译系统会自动选择您认为最适合您系统的并行任务计数。

`m`

如上所述，您可以通过在 `m` 命令行中列出相应名称来编译特定模块，而不是编译完整的设备映像。此外，`m` 还针对各种特殊目的提供了一些伪目标。以下是一些示例：

- **`droid-m droid`** 是正常版本。此目标在此处，因为默认目标需要名称。
- **`all-m all`** 会编译 `m droid` 编译的所有内容，加上不包含 `droid` 标记的所有内容。编译服务器会运行此命令，以确保包含在树中且包含 `Android.mk` 文件的所有元素都会编译。
- **`clean-m clean`** 会删除此配置的所有输出和中间文件。此内容与 `rm -rf out/` 相同。

运行 `m help` 即可查看 `m` 提供的其他伪目标。

开始运行！

您可以在模拟器上运行编译版本，也可以将其刷写到设备上。由于您已经使用 `lunch` 选择了编译目标，因此它不可能在编译目标以外的目标上运行。

注意： 请注意获取专有二进制文件，否则编译版本将无法在目标硬件上成功启动。有时，源代码可能会对不同编译版本和分支提供不同的二进制文件。如果此时获取二进制 Blob，则需要解压缩这些 Blob 并执行 `m installclean`，然后重新编译。要详细了解此过程，请参阅[下载源代码 \(/setup/build/downloading#obtaining-proprietary-binaries\)](#)一文。

使用 fastboot 刷机

要刷写设备，请使用 **fastboot**；在成功编译后，它应该会包含在您的路径中。如需相关说明，请参阅刷写设备 (</setup/build/running#flashing-a-device>)。

模拟 Android 设备

编译流程会自动将模拟器添加到您的路径中。要运行模拟器，请输入以下命令：

```
emulator
```

排查常见编译错误

Java 版本不正确

如果您尝试编译的 Android 版本与您的 Java 版本不一致，则 **make** 会中止，并显示如下消息：

```
*****
You are attempting to build with the incorrect version
of java.

Your version is: WRONG_VERSION.
The correct version is: RIGHT_VERSION.

Please follow the machine setup instructions at
  https://source.android.com/source/initializing.html
*****
```

以下是可能的原因和解决方案：

- 无法按照 [JDK 要求 \(/setup/build/requirements#jdk\)](/setup/build/requirements#jdk) 中指定的方式安装正确的 JDK。请确保您已按照[设置环境 \(/setup/build/building#initialize\)](/setup/build/building#initialize)和[选择目标 \(/setup/build/building#choose-a-target\)](/setup/build/building#choose-a-target)中的步骤操作。
- 之前安装的另一个 JDK 出现在您的路径中。将正确的 JDK 附加到路径开头，或者移除有问题的 JDK。

Python 版本 3

Repo 是基于 Python 2.x 中的特定功能构建的，与 Python 3 不兼容。要使用 Repo，请安装 Python 2.x：

```
apt-get install python
```

不区分大小写的文件系统

在 macOS 中的 HFS 文件系统上进行编译时，可能会遇到如下错误：

```
*****
You are building on a case-insensitive filesystem.
Please move your source tree to a case-sensitive filesystem.
*****
```

请按照创建区分大小写的磁盘映像 (</setup/build/initializing#creating-a-case-sensitive-disk-image>) 中的说明操作。

没有 USB 权限

默认情况下，在大多数 Linux 系统中，无特权的用户无法使用 USB 端口。如果您看到权限遭拒错误，请按照配置 USB 使用权限 (</setup/build/initializing#configuring-usb-access>) 中的说明操作。

如果上述规则设置完成后，adb 已在运行且无法连接到设备，您可以使用 `adb kill-server` 将其终止。此命令将使 adb 重启并采用新配置。

Content and code samples on this page are subject to the licenses described in the [Content License \(/license\)](/license).
Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-02-05.

