

刷写设备

本页提供了关于在特定设备上运行编译系统的详细信息，是对[编译准备工作 \(/setup/build/building.html\)](/setup/build/building.html)部分的补充。

编译 fastboot 和 adb

如果您还没有 **fastboot** 和 **adb**，则可以使用常规编译系统来进行编译。请按照[编译准备工作 \(/setup/build/building.html\)](/setup/build/building.html)中的说明操作，并将主 **make** 命令替换为以下命令：

```
make fastboot adb
```

启动进入 fastboot 模式

Fastboot 是一种引导加载程序模式，您可以在该模式下刷写设备。在设备冷启动过程中，可使用下文中的组合键进入 **fastboot** 模式。

您可以使用命令 **adb reboot bootloader** 直接重新引导至引导加载程序，也可以在冷启动期间使用以下某个组合键。

设备	代号	组合键
Pixel 3a XL	bonito	按住音量调低键，然后按住电源键。
Pixel 3a	sargo	按住音量调低键，然后按住电源键。
Pixel 3 XL	crosshatch	按住音量调低键，然后按住电源键。
Pixel 3	blueline	按住音量调低键，然后按住电源键。
Pixel 2 XL	taimen	按住音量调低键，然后按住电源键。
Pixel 2	walleye	按住音量调低键，然后按住电源键。
Pixel XL	marlin	按住音量调低键，然后按住电源键。
Pixel	sailfish	按住音量调低键，然后按住电源键。
hikey	hikey	连接 J15 的引脚 1-2 和 5-6。
Nexus 6P	angler	按住音量调低键，然后按住电源键。

设备	代号	组合键
Nexus 5X	bullhead	按住音量调低键，然后按住电源键。
Nexus 6	shamu	按住音量调低键，然后按住电源键。
Nexus Player	fugu	按住电源键。
Nexus 9	volantis	按住音量调低键，然后按住电源键。
Nexus 5	hammerhead	同时按住音量调高键和音量调低键，然后按住电源键。
Nexus 7	flo	按住音量调低键，然后按住电源键。
Nexus 7 3G	deb	按住音量调低键，然后按住电源键。
Nexus 10	manta	同时按住音量调高键和音量调低键，然后按住电源键。
Nexus 4	mako	按住音量调低键，然后按住电源键。
Nexus 7 (2012)	grouper	按住音量调低键，然后按住电源键。
Nexus 7 3G (2012)	tilapia	按住音量调低键，然后按住电源键。
Nexus Q	phantasm	启动设备，然后在 LED 指示灯亮起后用一只手盖住设备，直至指示灯变成红色。
Galaxy Nexus GSM	maguro	同时按住音量调高键和音量调低键，然后按住电源键。
Galaxy Nexus (Verizon)	toro	同时按住音量调高键和音量调低键，然后按住电源键。
Galaxy Nexus (Sprint)	toroplus	同时按住音量调高键和音量调低键，然后按住电源键。
Motorola Xoom	wingray	按住音量调低键，然后按住电源键。
Nexus S	crespo	按住音量调高键，然后按住电源键。
Nexus SG	crespo4g	按住音量调高键，然后按住电源键。

解锁引导加载程序

只有在引导加载程序允许的情况下，您才可以刷写自定义系统，而引导加载程序默认处于锁定状态。您可以解锁引导加载程序，但这样做会导致系统出于隐私原因而删除用户数据。解锁之后，系统会清空设备上的所有数据，即应用中的个人数据以及可通过 USB 访问的共享数据（包括照片和影片）。请先备份设备上的所有重要文件，然后再尝试解锁引导加载程序。

您只需解锁引导加载程序一次，并可视需要将其重新锁定。

解锁新款设备

自 2014 年以来发布的所有 Nexus 和 Pixel 设备（从 Nexus 6 和 Nexus 9 开始）都内置有恢复出厂设置保护功能，需要执行多个步骤才能解锁引导加载程序。

1. 要在设备上启用 OEM 解锁功能，请执行以下操作：
 - a. 在“设置”中，点按**关于手机**，然后点按版本号七 (7) 次。
 - b. 当看到“您已处于开发者模式”这条消息后，点按**返回**按钮。
 - c. 点按**开发者选项**，然后启用 OEM 解锁和 USB 调试。（如果 **OEM 解锁** 处于停用状态，请连接到互联网，以便设备可以至少检入一次。如果“OEM 解锁”仍处于停用状态，则说明设备可能已被运营商锁定 SIM 卡，系统无法解锁引导加载程序）。
2. 重新启动进入引导加载程序，然后使用 **fastboot** 解锁。

- 对于新款设备（2015 年及之后发布的设备）：

```
fastboot flashing unlock
```

- 对于老款设备（2014 年及之前发布的设备）：

```
fastboot oem unlock
```

3. 在屏幕上确认解锁。

注意：在 Nexus 10 上，解锁引导加载程序后，内部存储空间仍将保持未格式化状态。您可以依次使用 **fastboot format cache** 和 **fastboot format userdata** 来格式化设备。

重新锁定引导加载程序

要重新锁定引导加载程序，请执行以下命令：

- 对于新款设备（2015 年及之后发布的设备）：

```
fastboot flashing lock
```

- 对于老款设备（2014 年及之前发布的设备）：

```
fastboot oem lock
```

注意：在 Motorola Xoom 上重新锁定引导加载程序会清空所有用户数据（包括共享的 USB 数据）。

使用刷写锁定状态

`getFlashLockState()` 系统 API 会传输引导加载程序状

态，`PersistentDataBlockManager.getFlashLockState()` 系统 API 会返回兼容设备上引导加载程序的锁定状态。

返回值	条件
<code>FLASH_LOCK_UNKNOWN</code>	<p>仅升级到 Android 7.x 或更高版本且符合以下条件的设备会返回此值：设备支持刷写锁定/解锁功能，但之前不支持获取刷写锁定状态所需的引导加载程序变更。</p> <ul style="list-style-type: none"> • 运行 Android 7.x 或更高版本的新设备必须处于 <code>FLASH_LOCK_LOCKED</code> 或 <code>FLASH_LOCK_UNLOCKED</code> 状态。 • 升级到 Android 7.x 或更高版本但不支持刷写解锁/锁定功能的设备应返回 <code>FLASH_LOCK_LOCKED</code> 状态。
<code>FLASH_LOCK_LOCKED</code>	不支持刷写锁定/解锁的所有设备（即设备始终处于锁定状态）或支持刷写锁定/解锁且处于锁定状态的所有设备会返回此值。
<code>FLASH_LOCK_UNLOCKED</code>	支持刷写锁定/解锁且处于解锁状态的所有设备会返回此值。

制造商应测试由已锁定/解锁引导加载程序的设备返回的值。举例来说，Android 开源项目 (AOSP) 包含一个参考实现，该实现返回基于 `ro.boot.flash.locked` 引导属性的值。示例代码位于以下目录中：

- `frameworks/base/services/core/java/com/android/server/PersistentDataBlockManager`
- `frameworks/base/core/java/android/service/persistentdata/PersistentDataBlockManager`

选择设备编译系统

在 **lunch** 菜单中提供了建议的设备编译系统，在不使用任何参数的情况下运行 `命令` 即可查看。如需可用编译类型的信息及有关 **lunch** 命令的更多信息，请参阅[选择目标 \(/setup/build/building#choose-a-target\)](https://source.android.com/setup/build/building#choose-a-target)。

您可以从 developers.google.com (<https://developers.google.com/>) 下载 Nexus 设备的出厂映像和二进制文件。有关下载内容，请参阅[设备二进制文件 \(/setup/build/requirements#binaries\)](https://source.android.com/setup/build/requirements#binaries)。有关详情以及其他资源，请参阅[获取专有二进制文件 \(/setup/build/building.html#obtaining-proprietary-binaries\)](https://source.android.com/setup/build/building.html#obtaining-proprietary-binaries)。

设备	代号	编译配置
Pixel 3a XL	bonito	aosp_bonito-userdebug
Pixel 3a	sargo	aosp_sargo-userdebug
Pixel 3 XL	crosshatch	aosp_crosshatch-userdebug
Pixel 3	blueline	aosp_blueline-userdebug
Pixel 2 XL	taimen	aosp_taimen-userdebug
Pixel 2	walleye	aosp_walleye-userdebug
Pixel XL	marlin	aosp_marlin-userdebug
Pixel	sailfish	aosp_sailfish-userdebug
HiKey	hikey	hikey-userdebug
Nexus 6P	angler	aosp_angler-userdebug
Nexus 5X	bullhead	aosp_bullhead-userdebug
Nexus 6	shamu	aosp_shamu-userdebug
Nexus Player	fugu	aosp_fugu-userdebug
Nexus 9	volantis (flounder)	aosp_flounder-userdebug
Nexus 5 (GSM/LTE)	hammerhead	aosp_hammerhead-userdebug
Nexus 7 (WLAN)	razor (flo)	aosp_flo-userdebug
Nexus 7 (移动版)	razorg (deb)	aosp_deb-userdebug
Nexus 10	mantaray (manta)	full_manta-userdebug

设备	代号	编译配置
Nexus 4	occam (mako)	full_mako-userdebug
Nexus 7 (WLAN)	nakasi (grouper)	full_grouper-userdebug
Nexus 7 (移动版)	nakasig (tilapia)	full_tilapia-userdebug
Galaxy Nexus (GSM/HSPA+)	yakju (maguro)	full_maguro-userdebug
Galaxy Nexus (Verizon)	mysid (toro)	aosp_toro-userdebug
Galaxy Nexus (试验版)	mysidspr (toroplus)	aosp_toroplus-userdebug
Motorola Xoom (美国 WLAN 版)	wingray	full_wingray-userdebug
Nexus S	soju (crespo)	full_crespo-userdebug
Nexus S 4G	sojus (crespo4g)	full_crespo4g-userdebug

注意：请不要在原本搭载 Android 4.1.2 或更高版本的 Nexus 7 上使用 Android 4.1.1。

刷写设备

您可以通过运行一个命令来刷写整个 Android 系统；这样做可验证并确保正在被刷写的系统与已安装的引导加载程序和无线驱动程序兼容，还可以将启动、恢复和系统分区一起写入，然后重新启动系统。与 `fastboot oem unlock` 类似，刷写设备也会清空所有用户数据。

要刷写设备，请执行以下操作：

1. 在启动时按住相应的组合键或使用以下命令使设备进入 `fastboot` 模式：

```
adb reboot bootloader
```

2. 在设备处于 `fastboot` 模式后，运行以下命令：

```
fastboot flashall -w
```

`-w` 选项会清除设备上的 `/data` 分区；该选项在您第一次刷写特定设备时非常有用，但在其他情况下则没必要使用。

注意：在 Motorola Xoom 上使用 **fastboot** 创建的文件系统无法发挥最佳作用。建议您使用 `$ adb reboot recovery` 命令通过恢复模式重新创建文件系统。在恢复模式下，打开菜单（同时按**电源键和音量调高键**），擦除缓存分区，然后擦除数据。

使设备恢复到出厂状态

您可以在 [Nexus 和 Pixel 设备的出厂映像](https://developers.google.com/android/nexus/images) (https://developers.google.com/android/nexus/images) 中查看 Google 设备的出厂映像。Motorola Xoom 的出厂映像由 Motorola 直接分发。

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

Last updated 2020-02-05.

