

Huawei 手机 + ADB 的截屏/录屏方案（Deep Research）

Executive Summary

在部分华为/鸿蒙/EMUI 设备上，`screenrecord` 可能不可用或被裁剪，而 `screencap` 通常仍可用；因此“真机可感知证据”更稳妥的做法是：用 `UIAutomator` 的 `UiDevice.takeScreenshot()` 生成 PNG，再通过 `MediaStore` 写入 `Downloads`，最后在本机用 `ffmpeg` 合成 mp4，或者改用 `scrcpy` 在 PC 侧直接录屏 [1][2]。

本仓库已落地“帧截图→合成视频”的方案：instrumentation test 会把 `step-*.png` 写入 `Download/agent-browser-kotlin/e2e/frames/`（人类可见），本机运行 `scripts/pull-e2e-video.ps1` 会 `adb pull` 并生成 `adb_dumps/e2e/latest/e2e-latest.mp4`。

Key Findings

- `UiDevice.takeScreenshot(File)` 是官方 `UIAutomator` API，能把屏幕截图存成 PNG 并返回成功/失败布尔值 [1]。
- `UiDevice.executeShellCommand()` 被官方标注为“简单命令可用但不适合复杂命令/错误处理”，更推荐 `UiAutomation#executeShellCommand*` 或更可靠的替代路径 [3]。
- `scrcpy` 支持 PC 侧录屏（`--record file.mp4`），不依赖设备是否自带 `screenrecord`，在 OEM 定制系统上通常更稳 [2]。
- 设备侧 `screenrecord` / `screencap` 属于 shell 命令生态；如果设备缺少 `screenrecord`，可以用“截图帧 + 合成”或 `scrcpy` 作为录屏替代 [2][1]。

Detailed Analysis

1) 设备侧：screenrecord 可能缺失/不可用时怎么办

`screenrecord` 是 Android 生态里常见的录屏命令，但它是否存在、是否可用取决于具体 ROM 构建与厂商裁剪（例如你当前华为机上就实际观察到 `/system/bin/screenrecord` 不存在）。当 `screenrecord` 不可用时，常见替代有两条路线：

1. **PC 侧录屏 (scrcpy)**：把设备画面镜像到电脑并直接录制 mp4，最贴近“人类可感知证据”的诉求 [2]。
2. **设备侧截图帧 (PNG) → PC 合成 mp4**：每个关键 step 截一张图，能看到点击前后 UI 变化，且产物天然可 `adb pull`。

2) 截图：UIAutomator takeScreenshot vs 直接跑 screencap

在自动化测试里：

- `UiDevice.takeScreenshot(File)` 是 UIAutomator 直接暴露的 API，写文件简单且返回值明确 (true/false) [1]。
- `UiDevice.executeShellCommand()` 虽然存在，但官方明确提示它更适合简单命令，复杂命令 (引号/管道/错误处理) 不建议依赖 [3]。

因此，在“要稳定、要可回归、要证据”的前提下：

- 优先用 `takeScreenshot()` 生成 PNG (先写到 app cache/内部目录)。
- 再把 PNG 用 **MediaStore** 写入 **Downloads**，让人类可见、让 `adb pull` 简单。

3) 把截图写进 Downloads：为什么用 MediaStore

在 Android 10+ (API 29+) 的 scoped storage 下，直接写 `/sdcard/Download/...` 往往会被权限/存储策略卡住。MediaStore 是“写入公共媒体/下载目录”的标准路径：你创建的条目归你所有，写入不需要传统的外部存储写权限 (具体依 ROM 策略，但整体最符合现代 Android 的推荐方式)。

本仓库的 instrumentation 测试使用：

- `UiDevice.takeScreenshot(tmpFile)` 生成 PNG [1]
- `MediaStore.Downloads` + `RELATIVE_PATH="Download/agent-browser-kotlin/e2e/frames/"` 写入 Downloads

4) 录屏：scrcpy (推荐) vs 帧截图合成 (已落地)

scrcpy (更“像视频”)

scrcpy 官方支持录制功能，通过 `--record` 输出 mp4，并且不需要在设备上安装常驻应用 [2]。这在 OEM 定制 (华为/鸿蒙) 环境里常常比依赖设备自带的 `screenrecord` 更稳。

帧截图合成 (更“像证据”)

优点：

- 产物每一张都可直观看到步骤（非常适合 E2E 的 step-by-step 证据链）
- 不依赖 `screenrecord`

缺点：

- 视频是“定格帧串联”，不是连续帧率录屏

本仓库用 `ffmpeg` 把 `step-01..N.png` 按每张 3.5s 合成 mp4（可回放）。

Areas of Consensus

- UIAutomator 是 Android 官方推荐的 UI 自动化测试路线之一，适合做可回归的 E2E 证据 [1][4]。
- OEM 定制系统对 shell 命令/存储路径的行为差异很常见；在自动化链路上应偏向“更少依赖 ROM 特性”的方案（例如 `scrcpy` 或 `MediaStore`）[2][3]。

Areas of Debate

- 是否坚持“真录屏 mp4”：`scrcpy` 的录屏更像真实视频，但需要 PC 工具；帧截图合成不需要额外工具，但观感更像 step 证据而不是连续录屏。
- “截图写入 Downloads”的实现细节：不同 ROM 对 `scoped storage`、`Downloads` 的行为细节可能不同，必要时需要 fallback（例如写入 App 私有 `external files` 再用 `run-as` 导出）。

Sources

[1] Android Developers. “UiDevice (androidx.test.uiautomator) API Reference” — includes `takeScreenshot(File)` behavior and signature. (官方 API 文档，高可信)

<https://developer.android.com/reference/androidx/test/uiautomator/UiDevice>

[2] Genymobile. “scrcpy” (GitHub README) — documents recording and general behavior. (项目官方仓库，高可信)

<https://github.com/Genymobile/scrcpy>

[3] Android Developers. “UiAutomation API Reference” — documents `executeShellCommand()` and shell permission identity. (官方 API 文档，高可信)

<https://developer.android.com/reference/android/app/UiAutomation>

[4] Android Developers. “Write automated tests with UI Automator” — official guidance and positioning. (官方指南，高可信)

Gaps and Further Research

- 华为/鸿蒙具体版本对 `screenrecord` 缺失/禁用的范围与原因：需要按机型/版本收集样本（例如 `adb shell getprop ro.build.version.*` + `command -v screenrecord`）。
- 若未来必须“连续录屏”：优先评估 `scrcpy` 的安装与脚本化；其次评估 `MediaProjection`（需要一次性授权）并用 `UIAutomator` 自动点击授权弹窗。