

Unity StandAlone Patcher 使用说明

下载链接：

<https://security-patches.unity.cn/unity-patcher/UnityApplicationPatcher-1.0.7c-win64.zip>

<https://security-patches.unity.cn/unity-patcher/UnityApplicationPatcher-1.0.7c-macOS-Arm64.zip>

<https://security-patches.unity.cn/unity-patcher/UnityApplicationPatcher-1.0.7c-macOS-x64.zip>

使用说明

使用以下引擎版本打包的应用程序可能存在安全风险：

- Unity 2019.1 及之后的版本（mono 和 il2cpp）
- Unity 2017 及之后的版本（mono）
- 团结引擎（mono 和 il2cpp）

Unity 针对 2019.1 及更高版本以及团结引擎的编辑器发布了补丁版本。如果您的应用程序是使用下表中的引擎构建的，建议使用修补后的引擎编辑器重新构建应用程序。

Unity	2019.1 以上
Unity f1c1	2020.1 以上
团结	1.0 以上

然而，如果无法从源代码重新构建项目，或者项目是使用 2017/2018 版本构建的，也可以使用这里提供的 Patcher 工具来给应用程序打补丁，提高安全性。以下是 Patcher 工具对各个版本引擎构建出的应用程序的支持情况：

	Unity	Unity f1c1
UnityApplicationPatcher-1.0.6c-	Android, Win,	Android, Win*,

win64.zip	MacOS	MacOS*
UnityApplicationPatcher-1.0.6-macOS-Arm64.zip	Android, Win, MacOS	Android
UnityApplicationPatcher-1.0.6-macOS-x64.zip	Android, Win, MacOS	Android

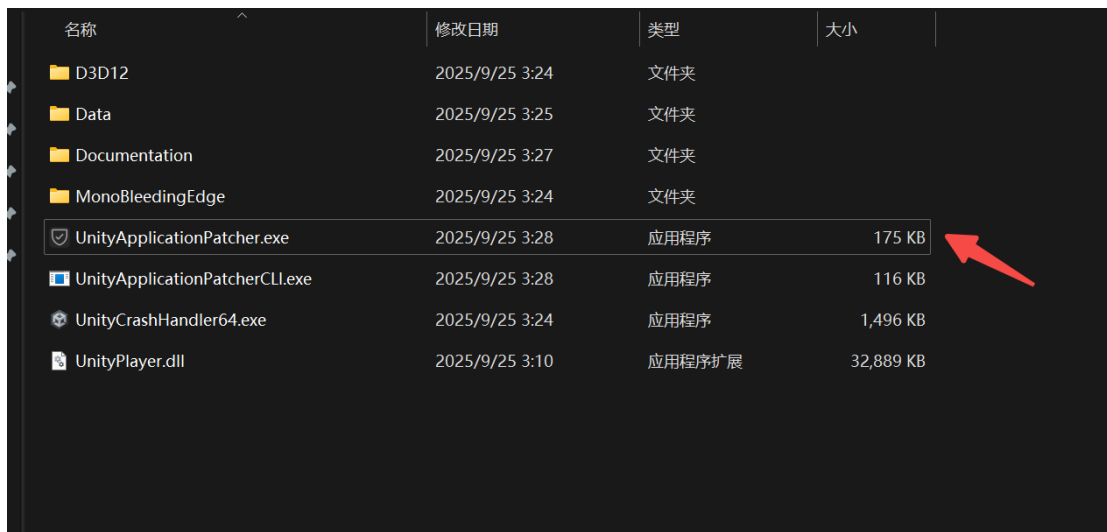
* 尚未完全覆盖 Windows 和 MacOS 平台的所有 f1c1 版本，正在不断完善中（无需重新下载 Patcher 工具，服务器会自动完善）

如果您使用的是团结引擎，请联系我们获取 Patcher 工具。

GUI 工具

Windows 平台

以 Windows 平台为例，双击启动 UnityApplicationPatcher.exe

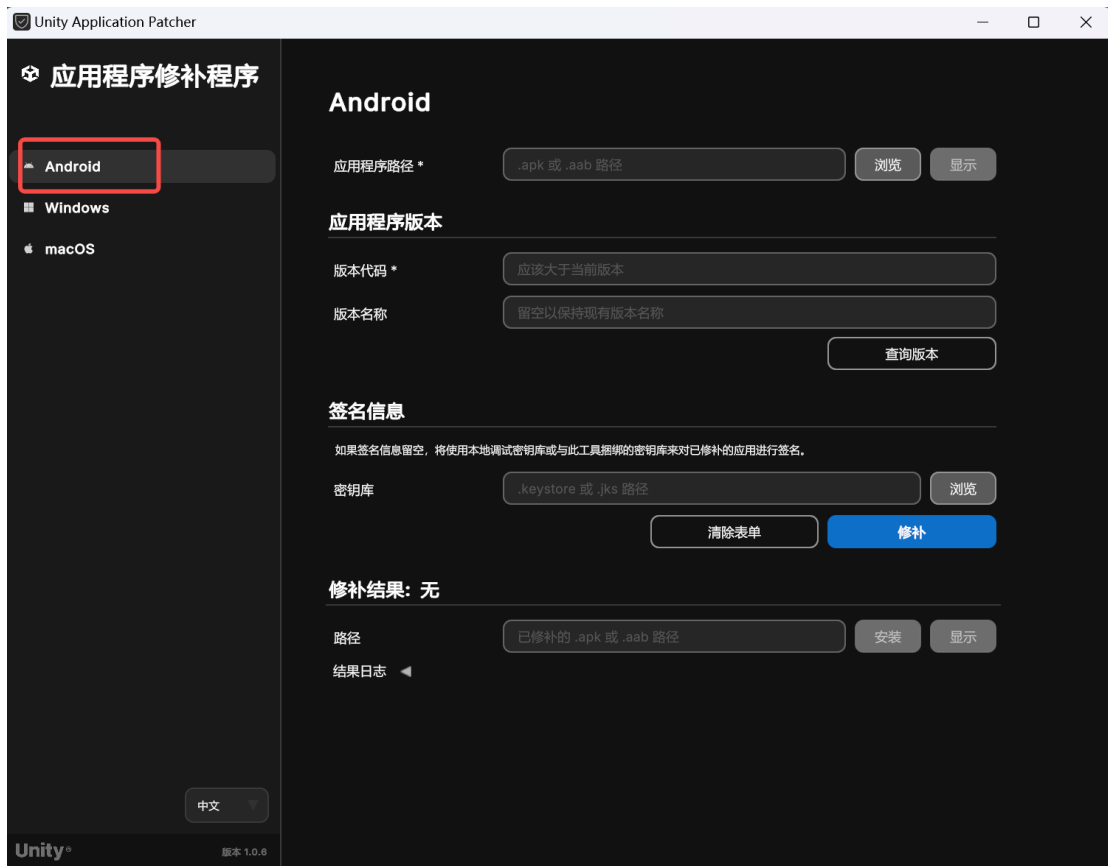


弹出 GUI 界面

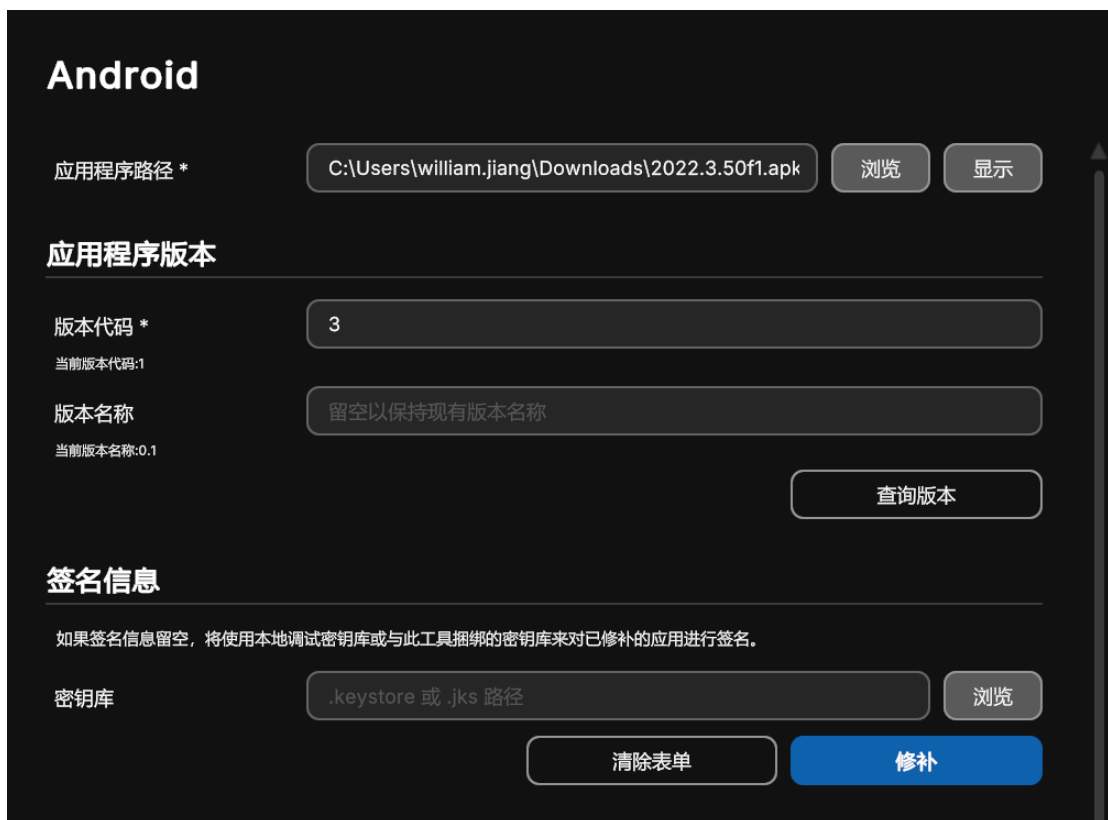
Patch APK

注意，当前发布的补丁工具暂不支持团结引擎打出的 APK 的修补

选择 Android 标签



填写 APK 路径，比原来大的版本号（点击查询版本可以查看当前 APK 版本号），和签名信息 keystore 文件（可选）



版本代码（version code）最好大于目前版本，默认值为 -1 即不修改 version code。

点击“修补”，执行完之后可以看到修补结果和产物路径

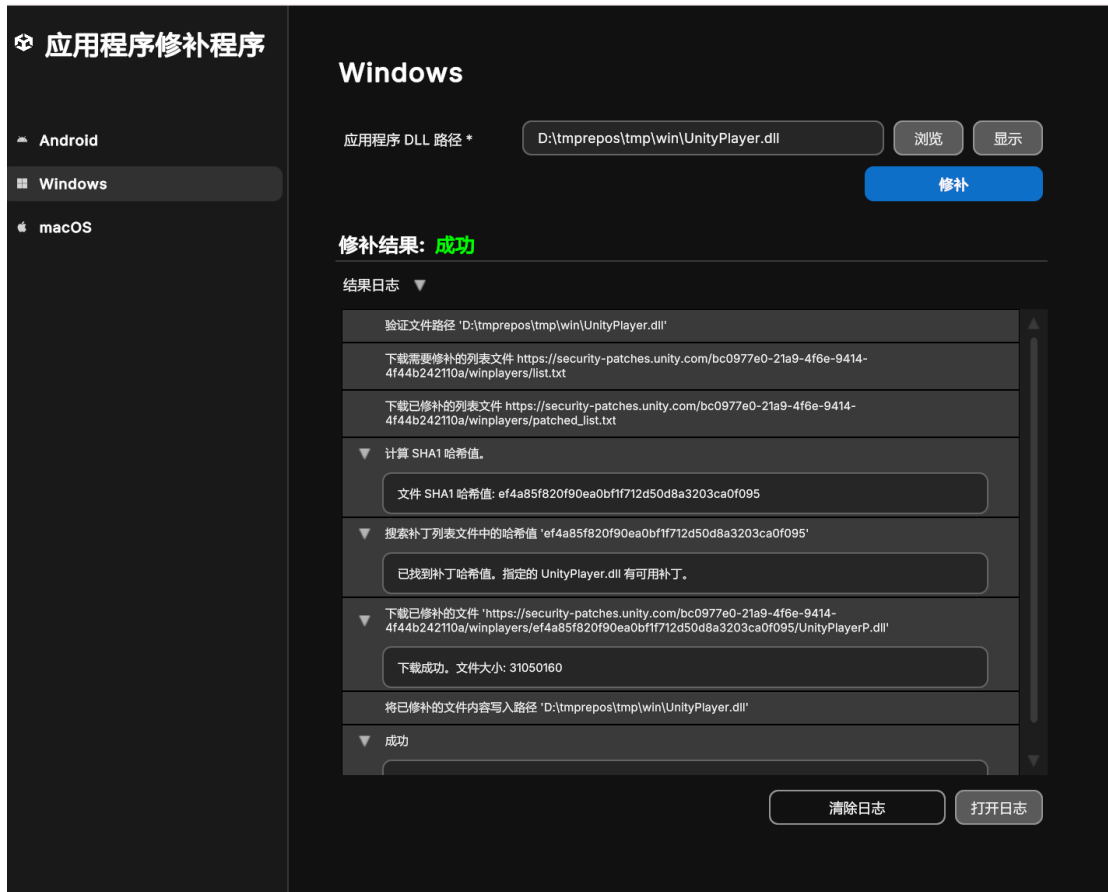


Patch Windows Player

注意，当前发布的 Windows 平台补丁工具暂不支持团结引擎打出的 Windows Player 的修补

注意：Windows Player 的修补方式会覆盖原始的 UnityPlayer.dll

1. 在 Windows 或 Mac 上启动应用程序。
2. 从侧边栏菜单中选择 **Windows** 按钮。
3. 导航到应用程序 DLL 路径栏，并选择浏览按钮。
4. 使用文件浏览器找到目标应用程序的 UnityPlayer.dll 文件。
5. 选择 UnityPlayer.dll 文件并在文件浏览器窗口中点击打开。
6. 按修补按钮。



Patch Mac Player

注意，当前发布的补丁工具暂不支持团结引擎打出的 **Mac Player** 的修补

注意，Windows Patcher 应用程序修补完的 Mac 应用程序，在 Mac 上直接运行可能失败，需要命令行执行 `chmod +x` 命令赋予执行权限。

1. 在 Windows 或 Mac 上启动应用程序。
2. 从侧边栏菜单中选择 **macOS** 按钮。
3. 导航到**应用程序路径**栏，并选择**浏览**按钮。
4. 使用文件浏览器找到应用程序捆绑包。例如：`Unity.app`。如果要修补专用服务器应用程序，请在服务器应用程序文件夹中找到 `UnityPlayer.dylib`。
5. 选择 `Unity.app` 或 `UnityPlayer.dylib` 文件并在文件浏览器窗口中点击打开。
 - a. **注意**：在 **Windows** 上使用此 UI 工具时，**浏览**功能仅支持 `UnityPlayer.dylib` 文件。这些文件位于 `myApplication.app/Contents/Frameworks/UnityPlayer.dylib`。还可以通

过在文本栏中输入应用程序的完整路径来修补 Unity.app。

6. 按修补按钮。

Mac 平台

与 Windows 平台操作类似，注意：当前 Mac 平台的补丁工具暂不支持 f1c1 版本和团结引擎的 Windows/Mac Player 产物修补。

Cli 命令行工具

命令行运行 UnityApplicationPatcherCLI 应用程序，使用 `-h` 参数可以看到支持的参数列表，按照指引即可

```
(base) PS D:\tmprepo\tmp\Build> .\UnityApplicationPatcherCLI.exe -h
UnityApplicationPatcher 1.0.6

Unity 应用程序修补程序

Unity 应用程序修补程序是一款提供 Android、Windows 和 macOS 应用程序修补的工具。修补 Android 应用程序、Windows 应用程序和 macOS 应用程序时，用法、选项和参数会有所不同。“修补 Android 参数”、“修补 Windows 参数”和“修补 macOS 参数”部分分别描述了每个平台的参数。

您可以在 Windows 或 macOS 上运行 Unity 应用程序修补程序。在 Windows 上运行此工具时使用 UnityApplicationPatcherCLI 来调用程序，而在 macOS 上使用 UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI 来调用程序。下面的“用法”部分对此进行了描述。

用法:
UnityApplicationPatcherCLI <android | windows> -applicationPath <路径/到/application.apk> -versionCode <代码> [-versionName <名称>] [-keystore <路径/到/密钥库>] [-keystorePassword <密码>] [-keyAlias <别名>] [-keyAliasPassword <密码>]
UnityApplicationPatcherCLI <windows> -applicationPath <路径/到/我的/application>
UnityApplicationPatcherCLI <windows> -unityPlayerLibrary <路径/到/我的/UnityPlayer.dll>
UnityApplicationPatcherCLI <macos> -applicationPath <路径/到/我的/application.app>
UnityApplicationPatcherCLI <macos> -unityPlayerLibrary <路径/到/我的/application.app/Contents/Frameworks/UnityPlayer.dylib>
UnityApplicationPatcherCLI <macos> -unityPlayerLibrary <路径/到/我的/application/UnityPlayer.dylib>

UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI <android | windows> -applicationPath <路径/到/application.apk> -versionCode <代码> [-versionName <名称>] [-keystore <路径/到/密钥库>] [-keystorePassword <密码>] [-keyAlias <别名>] [-keyAliasPassword <密码>]
UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI <windows> -applicationPath <路径/到/我的/application>
UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI <windows> -unityPlayerLibrary <路径/到/我的/UnityPlayer.dll>
UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI <macos> -applicationPath <路径/到/我的/application.app>
UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI <macos> -unityPlayerLibrary <路径/到/我的/application.app/Contents/Frameworks/UnityPlayer.dylib>
UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI <macos> -unityPlayerLibrary <路径/到/我的/application/UnityPlayer.dylib>
UnityApplicationPatcher.app/Contents/MacOS/UnityApplicationPatcherCLI <macos> -unityPlayerLibrary <路径/到/我的/application/UnityPlayer.dylib> [-allowStandaloneLibrary]

选项:
-android 指定修补 Android 应用程序。不能与 -windows 或 -macos 选项结合使用。
-windows 指定修补 Windows 应用程序。不能与 -android 或 -macos 选项结合使用。
-macos 指定修补 macOS 应用程序。不能与 -android 或 -windows 选项结合使用。
-v, [-verbose] 启用更详细的输出消息。
-quit 打印消息 -applicationPath 指定的 Android 应用程序的当前版本代码和版本名称。
-h 仅在 -android 选项时可用。需要和参数后，该工具会在不进行修补的情况下退出。

修补 Android 参数:
-applicationPath 需要修补的 Android 应用程序的路径。
-versionCode 修补后的应用程序的版本代码。必须高于当前应用程序版本。
-versionName 指向 Android APK/AAB 版本名称。与版本名称保持同步而版本名称 (可选)。
-keystore 或 .jks 文件的路径。如果省略此参数，则会使用本地测试密钥库或与此工具捆绑的密钥库来对修补后的应用程序进行签名 (可选)。
-keystorePassword 在 keystore 参数中提供的路径 -keystore 或 .jks 文件的路径 (可选)。
-keyAlias 在 keystore 参数中提供的关联 keystore 或 .jks 的唯一标识符 (可选)。
-keyAliasPassword 在 keyAlias 参数中提供的密钥名称的密码 (可选)。

修补 Windows 参数:
-applicationPath 需要修补的 Windows 应用程序的路径。
-unityPlayerLibrary 需要修补的 Windows 应用程序 UnityPlayer.dll 的路径。

修补 macOS 参数:
-applicationPath 需要修补的 macOS 应用程序的路径。
-unityPlayerLibrary 需要修补的 macOS 应用程序 UnityPlayer.dylib 的路径。
-allowStandaloneLibrary 允许通过 -unityPlayerLibrary 指定一个不包含在应用程序内的独立的 UnityPlayer.dylib 文件。默认情况下，这是不允许的，因为如果 Unity 应用程序修补程序找不到应用程序文件，则无法修补应用程序文件中需要修补的任何其他文件。
```

使用 `-verbose` 或 `-v` 参数获取更详细的修补过程日志。

使用 `-versionCode`，指定 patch 之后 APK 的 version code 版本号，建议大于目前版本。如果不指定，默认使用当前 version code 版本号。

应用程序返回代码

Unity 应用程序修补程序退出时返回指定的代码，以指示其修补操作结果。这些代码对于自动化、脚本编写和故障排除特别有用。

代码	描述	场景
0	成功	已成功应用补丁或已成功运行 help 命令。
1	修补失败（通用）	修补操作因故失败。
2	未找到补丁（失败）	未找到该二进制文件的补丁。
3	捕获异常	修补过程中捕获到异常。
64	命令行参数无效	收到无效的命令行参数。请参阅上面的命令行参数。
183	不需要补丁（已应用）	无法应用补丁，但被视为成功（已应用或不需要补丁）。

点击图片可查看完整电子表格

注意事项：

- 如果应用程序使用了防篡改或反作弊方案，以上方法将无法使用，需要使用修复后的引擎版本重新打包。
- 使用以上方法修复，将影响使用 `-xrsdk-pre-init-library` 和 `-overrideMonoSearchPath` 命令行参数。
- Patch 完成后，请验证应用的功能完整性。
- 如果 `UnityApplicationPatcher` 的路径长度（不包括 `UnityApplicationPatcher`）在 macOS 上超过 174 个字符，或在 Windows 上超过 193 个字符，则 `UnityApplicationPatcher` 工具可能无法启动，或显示异常，或修补失败。
- 在 Windows 上通过命令行工具使用 Unity 应用程序修补程序时，非英文字符可能无法在命令提示符窗口中正确显示。更改命令提示符字体（例如更改为 `SimSun-ExtB`）可以解决此问题。要更改字体，请右键点击命令提示符窗口的顶栏，选择**属性**，点击**字体**选项卡，然后选择新字体。点击**确定**应用所做的更改。
- 如果遇到本文档中没有提到的问题，请参考压缩包中的 `Documentation` 目录或者咨询引擎支持人员。
- Windows Patcher 应用程序修补完的 Mac 应用程序，在 Mac 上直接运行可能失败，需要命令行执行 `chmod +x` 命令赋予执行权限。
- Patch Android APK/AAB 时，版本代码（version code）最好大于目前版本，默认值为 `-1` 即不修改 version code。