

Python 工具 Win7 相容打包

Python 3.8-32 + UCRT 內附 — 讓 PyInstaller 打包的 EXE 在 Win7 / 32 位元 Windows 都能執行 | 2026-09-19

一、訴求與現象

客戶工作站仍有 Win7（含 32 位元版）。在 Win7 上開啟 INI 設定編輯器 INI_Editor.exe，系統直接跳出錯誤：

INI_Editor.exe - 系統錯誤
無法啟動程式，因為您的電腦遺失
api-ms-win-core-path-l1-1-0.dll。請嘗試重新安裝以修正這個問題。

程式連第一行都還沒執行就被 Windows 擋下，所以不會有任何 log。盤點工具目錄後發現不只這一支，共有 8 支會派送到客戶端的 Python 程式有同樣問題，分成兩類：

工具	修正前	在 Win7 / 32 位元的結果
INI_Editor	Python 3.12 / 32 位元	Win7 缺 api-ms-win-core-path 起不來
check_system_ini	Python 3.12 / 32 位元	同上
go2ini	Python 3.12 / 32 位元	同上
chk_dupacc	Python 3.14 / 64 位元	32 位元 Windows 完全不能執行（不是有效的 Win32 應用程式）
chk_headbody	Python 3.14 / 64 位元	同上
prg_indent	Python 3.14 / 64 位元	同上
stk_updater WatchDog_Stk Updater	Python 3.12 / 32 位元	Win7 缺 api-ms-win-core-path 起不來（客戶主機也可能是 Win7）

三個派送目錄（toAWI_Home\tools
x2、update_pack\TOOLS）裡的版本都一樣，共 18
個檔案需要重打。更新程式 stk_updater 和它的看門狗
WatchDog_StkUpdater 部署在 E:\update\Update 和

update_pack\SOCKET，共 4 個檔案。原本以為它們只會在 Win10 的客戶主機執行，所以一直用 Py3.12，但實際上客戶主機也可能是 Win7。

二、根本原因

- Python 3.9 起官方放棄 Win7。3.9+ 的 python3XX.dll 啟動時需要 PathCch 系列 API（放在 api-ms-win-core-path-l1-1-0.dll），這個 DLL 只有 Win8 以後才有。Python 3.8.10 是最後一版支援 Win7 的 Python，所以 3.12 打包出來的 EXE 在 Win7 上一定起不來，跟 32/64 位元無關。
- EXE 位元數由「執行 PyInstaller 的那支 python」決定，spec 裡的 target_arch=None 不會綁定位元數。chk_dupacc 這三支的打包 bat 寫的是不帶路徑的 python，PATH 找到的是 C:\Python314（64 位元），於是打出 64 位元 EXE。
- Win7 沒有內建 UCRT（通用 C 執行階段，Win10 才內建）。Python 3.8 依賴 UCRT，所以打包時要把 UCRT DLL 一起放進 EXE（app-local）。

三、踩坑與陷阱

陷阱	症狀 / 說明	對策
以為「打成 32 位元」就能在 Win7 執行	INI_Editor 早就是 32 位元，Win7 照樣起不來：位元數是對的，Python 版本不對	Win7 相容要同時符合「32 位元」和「Python 3.8」
打包 bat 寫不帶路徑的 python	依 PATH 找到 C:\Python314 64 位元，公司機能跑、客戶 32 位元機完全不動	一律寫死 E:\Python38-32\python.exe
Win7 缺 UCRT	即使用 Python 3.8，沒裝 KB2999226 的 Win7 仍缺 api-ms-win-crt-*.dll	spec 的 binaries 加入 Windows SDK Redist\ucrt\DLLs\x86*.dll (10.0.10240 app-local 版本，Win7 SP1 以上可用)
Win7 缺 KB3063858	Python 3.8 使用新的 DLL 載入 API，沒補這個更新時會出現「_socket 參數錯誤」	客戶 Win7 補裝 KB3063858（它取代已下架的 KB2533623）
PyInstaller 5.13 不認 optimize=0	Python 3.12 產生的 spec 在 Analysis() 裡有 optimize=0，舊版 PyInstaller 會報錯	py38 用的 spec 拿掉這個參數

陷阱	症狀 / 說明	對策
用 DLL 名稱判斷相容性會誤判	onefile 會壓縮內容，api-ms-win-core-path 字串不一定以明文出現在 EXE 裡	改用 EXE 裡的 python3XX.dll 字串判斷：python38 表示相容；UCRT 則看 api-ms-win-crt-runtime 字串
spec 目錄和 .py 不在同一個磁碟	PyInstaller 報 ValueError: path is on mount E:, start on mount C:	specpath / workpath 放在和原始碼同一個磁碟
用 Python 寫入 bat 路徑時跳脫字元出錯	b"...Kits\10" 被解讀成 \1 八進位、"\x86" 被解讀成 0x86 位元組，bat 裡混入控制字元	用 bytes([92]) 組反斜線，寫完後檢查檔案內沒有 < 0x20 的控制字元

四、解決方法

- 新增 HBSTK\build\py38spec*_py38.spec (6 支)，每支 spec 都附上 UCRT x86 DLL，並拿掉 optimize=0。
- 新增統一打包腳本 HBSTK\build\Build_Py38_Tools.bat：固定使用 E:\Python38-32，逐支打包後自動覆蓋 3 個派送目錄裡的同名 EXE（只覆蓋原本就存在的檔案，不會新增檔案）。
- 修正舊腳本，避免下次重打時退回錯誤版本：Build_Deploy_INI_Editor.bat（改用 Python38-32 + py38 spec）、doc\tools_build_recall.bat（原本用不帶路徑的 python，改成 Python38-32 + --add-binary 附帶 UCRT）。
- stk_updater 與 WatchDog：新增 py38spec\stk_updater_py38.spec（保留 collect_all("colorama")，另外加上 UCRT；Py38 要先 pip install colorama）與 WatchDog_StkUpdater_py38.spec；update_server\Build_Deploy.bat 改用 Python38-32，並同步更新 update_pack\SOCKET 裡的同名 EXE。

驗證結果：派送目錄內 18 個 EXE 全部是 python38 / x86 (0x14C) / 含 UCRT；5 支主控台工具在本機實際執行時都能正常進入程式邏輯，INI_Editor 的 GUI 可以正常啟動；stk_updater 共 4 個檔案同樣通過檢查，--colortest（不跑更新流程）實際執行時上色與中文輸出都正常。以 vermin 靜態分析 8 支原始碼，最低需求為 Python 3.7，確認 3.8 可用。

五、使用技術與用法

1. 重新打包並派送（全部或單支）

```
E:\ConvertFW\HBSTK\build\Build_Py38_Tools.bat (all 6 tools)
E:\ConvertFW\HBSTK\build\Build_Py38_Tools.bat go2ini (one tool)
```

2. Win7 相容 spec 的關鍵兩行（其餘與一般 spec 相同）

```
binaries=[(str(_p), '.') for _p in __import__('glob').glob(
    'C:/Program Files (x86)/Windows Kits/10/Redist/ucrt/DLLs/x86/*.dll')],
# no optimize=0 (PyInstaller 5.13)
```

命令列 `--onefile` 打包的等效寫法：

```
"E:\Python38-32\python.exe" -m PyInstaller --onefile --console ^
--add-binary ^
"C:\Program Files (x86)\Windows Kits\10\Redist\ucrt\DLLs\x86\*.dll;." ^
xxx.py
```

3. 打包後的檢查（三項都要符合）

```
import re, struct
d = open('xxx.exe', 'rb').read()
pe = struct.unpack_from('<I', d, 0x3c)[0]
print(hex(struct.unpack_from('<H', d, pe+4)[0])) # 0x14c = 32-bit OK
print(set(re.findall(rb'python3\d+\.\dll', d))) # python38 = Win7 OK
print(b'api-ms-win-crt-runtime-l1-1-0.dll' in d) # True = UCRT bundled
```

4. 新增 Python 工具時的判斷原則

元件會在這裡執行	Python 版本	位元
工作站（可能是 Win7）：pusher、session_pusher、ws_beacon、img2text、prn2xlsx、stk_updater、WatchDog_StkUpdater、INI_Editor、check_systemini、go2ini、chk_dupacc、chk_headbody、prg_indent	3.8-32 + UCRT	32
只在中繼伺服器或公司桌機（Win10）執行：server_updater、relay_server、Commander 等	3.12-32 可以	32

5. 客戶端 Win7 前置條件：Win7 SP1+KB3063858。UCRT 已經包在 EXE 裡，不需要另外安裝。