

STK Agent 生存加固 + 配套改進

Agent 永不死 / 自動部署 / 雙向同步 / 錯誤回報強化
2026-05-29

一、整體訴求

客戶 Agent 系統「時間久了就死亡」、「watchdog 也失效」、「PUSH_AND_UNZIP 失敗」。使用者「我這台改的 PrnAsk 設定被 SyncNow 覆蓋」、「200% DPI 螢幕截圖只有 1/4 大小」、「7 個工作站的錯誤全跑到 S001 資料夾」。

Agent 是整個 SOCKET 系統的核心 — Agent 死，自動錯誤回報死、ZIP 部署死、Commander 看不到、SHUTDOWN 失效。今天加倍用心，**從根上把 Agent 弄到永不死**。

二、完成項目總覽

模組	改進	影響
Agent main loop	外層 BEGIN SEQUENCE/RECOVER 包整個 ListenLoop	crash 後 5 秒自動重啟，Agent 永不死
Agent main loop	ExecuteCommandRelay 也包 SEQUENCE	cmd 處理 crash 不影響主 loop
Agent dead-detect	90 秒沒收到任何資料 → 強制 reconnect	解 TCP half-open zombie
watchdog_agent.bat	/dev/null → nul (13 處)，%~dp0 不依賴 HKCU	watchdog 真的有效
install_agent.bat	LOG + errorlevel 檢查 (netsh/schtasks/copy)	事後可追查為何裝失敗
install_agent_silent.bat	重寫: chcp 950 + LOG + station 優先順序	不再裝錯 station
Agent CheckLocalInstall	30s 掃 I:\tools\ ZIP 自解到本機 C:\	不靠 TCP push，工作站自助
stk_updater.py	LIST_AGENTS 寬鬆 filter + 推 ZIP 到 D:\images\tools	空 KMSN agent 也納入；ZIP 落到共享給 Agent 自取

模組	改進	影響
Agent bidirectional sync	INI 雙向 newest-wins; content-first 防 ping-pong	使用者改 Preview 自動推上 W:\, 不再被覆
Agent CheckIniSync	30s 自動觸發 sync	不必按 SyncNow 也會推
HBSTK ERRORSYS DPI	PowerShell base64 + SetProcessDPIAware()	200% DPI 螢幕截全圖, 不再只 1/4
HBSTK ERRORSYS ARGS	加 ARGS=[1]type:val [2]... 到 trigger	錯誤現場 e:args 一覽, debug 不靠猜
Agent ProcessErrorPending	FTP 路徑優先用 COMPUTER 而非 STATION	7 工作站分流正確

三、Agent 生存防護五層

把 Agent 視為「永不死的服務」，需要多層防護。任何一層失效，下一層接上。

層	機制	位置	防什麼
L1.A	ExecuteCommandRelay BEGIN SEQUENCE/RECOVER	stk_agent.prg 主 loop 內	cmd 處理拋例外
L1.B	Main 外層 SEQUENCE 包整個 ListenLoop	stk_agent.prg Main()	ListenLoop 任何位置例外
L2	外層 crash → 5 秒後重啟 ListenLoop (無限)	Main DO WHILE .T.	process 不退出，自我復活
L3	90 秒沒收到任何資料 → 強制 reconnect	ListenLoop timeout 段	TCP half-open zombie
L4	30 秒寫一次 .heartbeat 檔	ListenLoop 開頭	給 watchdog 判 stale
L5.A	watchdog %~dp0 (不靠 HKCU)	watchdog_agent.bat	Task Scheduler SYSTEM 跑也能找到
L5.B	watchdog tasklist/taskkill/timeout 用 nul	watchdog_agent.bat (13 處)	redirect 真的生效

關鍵原始碼示範 (L1.B + L2)：

```
DO WHILE .T.
  lNormalExit := .F.
  BEGIN SEQUENCE WITH {|e| Break(e)}
  RelayListenLoop()
  lNormalExit := .T. /* 走到這 = 正常 tray exit */
  RECOVER USING oErr
  LogMsg("FATAL CRASH: " + oErr:description + " - restarting in 5s")
  hb_idleSleep(5)
  END SEQUENCE
  IF lNormalExit ; EXIT ; ENDIF /* 只有正常退出才真離開 */
ENDDO
```

四、踩坑清單 + 解法

坑 1: AgentErrorHandler BREAK 找不到 SEQUENCE → process 直接 crash

症狀：Agent 在系統匣可見但 Relay 看不到、SHUTDOWN 沒回應。

原因：global ErrorBlock 設了 AgentErrorHandler，handler 做 BREAK(e)。但主 loop 沒有 BEGIN SEQUENCE，BREAK 找不到地方落腳 → process 終止。

解法：Main 加外層 SEQUENCE 包 RelayListenLoop。crash 後 RECOVER → log → sleep 5s → restart。

坑 2: TCP half-open 偵測不到

症狀：Agent 正常跑但 Relay 認為 connection dead，雙方都不知道對方斷了。

原因：RelayKeepAlive 只 send 不等 ACK。網路靜默斷時，TCP send 進 buffer 看起來 OK，但 Relay 收不到。Relay 90 秒 timeout 後關 connection，Agent 還以為連著。

解法：追蹤 nLastRecv（最後收到資料的時間）。若 90 秒沒收到任何東西（包含 KEEPALIVE_ACK 或 cmd），視為死連線，hb_socketClose + s_hRelay := NIL → 下次迴圈會重連。

坑 3: watchdog_agent.bat 用 /dev/null 從來沒生效

症狀：watchdog 排程跑了但 Agent 還是死的，沒重啟。

原因：BAT 裡面 13 處用 /dev/null 重定向（Unix 語法）。Windows 把它當 literal 檔名，創出怪檔，redirect 失敗，errorlevel 不對，邏輯整個跑歪。

解法：全改 nul（Windows 標準 null device）。同時 watchdog 不再讀 HKCU，改用 %~dp0（BAT 自身位置）找 Agent EXE，避開 Task Scheduler SYSTEM 看不到 user HKCU 的問題。

坑 4: install_agent.bat 沒 errorlevel 檢查

症狀：客戶機 install 完看似正常，但 Windows Firewall 沒規則，TCP 9529 不通。

原因：netsh / schtasks / copy 都沒檢查 errorlevel，失敗也沒人知道。

解法：每個關鍵步驟後 if errorlevel 1 (echo [FAIL] ...) else (echo [OK] ...) >> LOG_FILE。LOG 寫到 W:\socket\install_log_{COMPUTERNAME}.txt，下次客戶有問題立刻看 log 知道哪步失敗。

坑 5: TCP push 失敗 (WinError 10060)

症狀：客戶 7 工作站，4 台 ZIP 推不過去。

原因（查證後）：WinError 10060 = TCP SYN 送出無回應 → Windows Firewall silent drop。確認那 4 台工作站防火牆沒裝 STK Agent TCP 9529 規則。原 install_agent.bat 沒檢查，所以無從追溯。

解法：(a) 加 install log 下次裝完即知 (b) 不靠 TCP 推 — Agent 自己每 30 秒從 I:\tools\ 拉 ZIP，繞過防火牆。Updater 把 ZIP 推到 D:\images\tools (= 工作站視角 I:\tools)。

坑 6: SyncNow 把使用者改的 INI 覆蓋掉

症狀：使用者在 PrnAsk 改 Preview 縮放，存到 C:\STK\S00X\system.ini。下次 Commander SyncNow → W:\system.ini 覆蓋本機 → 設定白費。

解法：SyncFilesFromNetwork() 改成「.exe 單向 W:\→本機；.ini 雙向 newest-wins」。使用者改 → local 較新 → PUSH 上 W:\。管理員改 INI Editor → W:\ 較新 → PULL 下 local。

另加 CheckIniSync() 每 30 秒自動觸發，不必按 SyncNow 也會推。

坑 7: bidirectional sync 無限 ping-pong

症狀：剛做完坑 6 修法，log 顯示每 30 秒 PUSH/PULL 交替，永不停。

原因：hb_MemoWrit 寫檔時，目標檔 mtime = 寫入當下 = NOW。PUSH 完後 W:\ mtime 比 local 新。下次比較 W:\ 較新 → PULL local。PULL 完 local mtime 變 NOW → 下次比較 local 較新 → PUSH...

解法：內容優先比較。兩端 hb_MemoRead 比對字串，相同就跳過（不管 mtime）。內容不同才比 mtime 決定方向。PUSH 之後兩端內容一致 → 下次 check skip → 不再 flap。

坑 8: PowerShell 截圖 DPI 200% 變 1/4

症狀：200% DPI 螢幕的工作站，DefError 截到的 BMP 只有左上 1/4。

原因：PowerShell 預設 DPI-unaware。PrimaryScreen.Bounds 回邏輯像素 (200% 下 1920x1080 變 960x540)。CopyFromScreen(0, 0, 0, 0, 960, 540) 只抓螢幕左上 960x540 區域。

解法：PowerShell 開頭 Add-Type 宣告 user32!SetProcessDPIAware，呼叫一次，再做截圖 → Bounds 回物理像素。為了避開 cmd 雙引號跟 "user32.dll" 內嵌的 quoting

地雷，整段 PowerShell 用 base64 encode 後傳給 powershell -EncodedCommand。

坑 9: 7 工作站錯誤全傳到 S001 (FTP)

症狀：S007 用戶觸發的錯誤，FTP 上跑到 /H/stk-error/{KMSN}/S001/。

原因：trigger 的 STATION 欄位用 MEMVAR->cTopWare 推導。但客戶 7 工作站共用同一份 stk.ini 設定，cTopWare 都是 1 → STATION 都寫 S001 → 全擠 S001 資料夾。

解法：Agent ProcessErrorPending() 的 FTP 路徑優先用 trigger 內 COMPUTER 欄位 (= GetEnv("COMPUTERNAME")), 因為客戶機名就是 S007/S005，唯一識別。COMPUTER 缺才 fallback STATION，再缺才 fallback Agent 自己的 s_cStation。

五、檔案修改清單

檔案	修改內容
SOCKET/StkAgent/source/stk_agent.prg	Main 外層 SEQUENCE; ExecuteCommandRelay SEQUENCE; 90s dead-detect; CheckLocalInstall (I:\tools\); CheckIniSync; SyncFilesFromNetwork bidirectional newest-wins + content-first; ProcessErrorPending FTP 路徑優先用 COMPUTER
SOCKET/StkAgent/build/watchdog_agent.bat	/dev/null → nul (13 處); 用 %~dp0 取代 HKCU registry lookup
SOCKET/StkAgent/build/install_agent.bat	加 LOG + errorlevel (INSTALL + UPDATE 兩段); AGENT_PATH fallback StkAgentWD
SOCKET/StkAgent/build/install_agent_silent.bat	重寫: chcp 950 + LOG + errorlevel; station 偵測優先 COMPUTERNAME 再 current dir 再 scan
HBSTK/build/update_server/stk_updater.py	LIST_AGENTS 診斷 log; KMSN filter 寬鬆 (空 kmsn 納入); shared_tools_dirs 支援多路徑 (D:\images\tools + I:\tools); copy ZIP to shared
HBSTK/source/ERRORSYS.PRG	BuildScreenshotCmd 用 base64 + DPI-aware; GetErrArgsStr 助手; trigger 加 ARGS= 欄位

六、教訓 (給未來踩同坑的人)

#	教訓
1	global ErrorBlock 做 BREAK(e) 必須有外層 SEQUENCE 接住, 否則 process 直接死
2	TCP 不主動 keepalive (with ACK) 偵測不到 half-open, 必須業務層追蹤 nLastRecv
3	Windows BAT 絕對不能用 /dev/null, 用 nul
4	Task Scheduler 跑 SYSTEM 看不到 user HKCU, 腳本應用 %~dp0 找 path
5	netsh/schtasks/copy 默默成敗, 必須 errorlevel 檢查並 log
6	雙向 sync 用 mtime 必然 ping-pong; 改成內容先比對才穩
7	PowerShell 截圖預設 DPI-unaware; 要 SetProcessDPIAware() 才拿物理像素
8	PowerShell 嵌入 C# DllImport("user32.dll") 跟 cmd quote 打架, 用 base64 encode 跳過

#	教訓
9	客戶共用 stk.ini → cTopWare 不可靠；應該用 COMPUTERNAME 唯一識別工作站
10	install 失敗最大成本是無 log；任何安裝/部署腳本一定要寫 log