

UpdStk 更新程式 — 計數器溢位修正報告

日期：2026-09-04 對象：UpdJson32.exe / UpdStk32.exe

一、訴求與現象

客戶反映：UpdStk 更新程式執行時，畫面中央進度框（ProcessBox）顯示的筆數計數器只能計到

999999；一旦處理的資料筆數超過一百萬，計數器就溢位顯示成「*****」（六個星號），無法顯示實際筆數。

需求：把計數器上限由 6 位（999999）擴大為 7 位（9999999），可容納千萬級筆數。

二、踩坑與陷阱

1. 『計數器在哪裡』的誤判：一開始以為計數器由 ProcessBox() 函式內部維護。實際閱讀後發現 ProcessBox() 只負責顯示傳入的訊息字串，本身沒有計數器；真正的計數器是各呼叫端用 StrZero(++nI, 6) 產生（第二參數 6＝欄寬 6 位）再組進訊息，超過六位即溢位成星號。
2. 一檔兩用：UPDSTK.PRG 同時被 UpdJson32 與 UpdStk32 兩個 EXE 引用，改一次要重編兩個 EXE。
3. 同一行的其他計數不可誤傷：AppendCounter 那行還有 StrZero(nAlias, 3) / StrZero(nTotal, 3)（第幾個表／共幾個表），此為 3 位、永遠不會破百萬，必須保留不動，只改筆數的那一個。
4. BIG5
編碼陷阱：這些行都含中文訊息（BIG5）。若用一般編輯器改易混入 UTF-8 造成整行亂碼，故一律以 Python bytes-only 精確替換 ASCII 片段「, 6)」→「, 7)」，完全不觸碰中文位元組。

三、解決方法

將所有「資料筆數」計數器的欄寬由 6 位改為 7 位（StrZero(…, 6) → StrZero(…, 7)），共 5 處；表數計數(, 3)維持不變。改後未破百萬者僅多顯示一個前導

0，無副作用。

檔案	修改處	用途
UPDSTK.PRG	StrZero(++nI, 6) → 7	檔案更新筆數（第 1589 行）
UPDSTK.PRG	StrZero(nAppend++, 6) → 7	匯入新結構筆數（AppendCounter，第 2065 行）
UpdateByJson.PRG	StrZero(++nI, 6) → 7	消掉舊筆（第 397 行）
UpdateByJson.PRG	StrZero(++nI, 6) → 7	消去筆數（第 419 行）
UpdateByJson.PRG	StrZero(++nI, 6) → 7	更新運算筆數（第 928 行）

四、使用技術與用法

- 定位：以 grep 搜尋 StrZero(…, 6) 找出所有 6 位計數器，並確認 ProcessBox() 本身無內部計數器。

- 修改：Python bytes-only

讀寫檔案，精確替換「StrZero(++nI, 6)」 「StrZero(nAppend++, 6)」，不做 decode/encode 循環，保護 BIG5 中文。

- 編譯：以 hbm2 分別重建兩個 EXE：

hbm2 updjson32.hbp -comp=bcc → UpdJson32.exe

hbm2 updstk32.hbp -comp=bcc → UpdStk32.exe

- 產出：UpdJson32.exe（2,145,280 bytes）、UpdStk32.exe（2,114,048 bytes），位於 E:\ConvertFW\UpdStk\build\。

- 備份：UPDSTK.PRG.bak_counter7_20260904、UpdateByJson.PRG.bak_counter7_20260904。