

SetColor 恢復修正 工作報告

STK 庫存系統 — print_spool.prg

日期：2026-04-22

1. 訴求與現象

進入主選單後，透過右鍵叫出系統選單，點選「直接/預覽列印切換」(key 9997) 後，主選單的高亮色彩跑掉 (變成 W+/B 白底亮字)，恢復不了正常配色。

同樣地，點選「外部印表機列印切換」(key 9998) 後也有類似問題。

根本原因：SetColor 被改為 W+/B 之後，沒有在所有路徑正確恢復。

2. 根因分析

2.1 StkTogglePntGoFmt() — 主 Bug

[BUG] cOldColor 宣告為 LOCAL 但從未賦值 SetColor()，呼叫 SetColor(cOldColor) 等同 no-op

函式宣告：

```
FUNCTION StkTogglePntGoFmt()
  LOCAL cIni := hb_DirBase() + "System.ini"
  LOCAL cContent, nPos, nEnd, nNew, cScreen, cOldColor ← 宣告但未賦值
  ...
  cScreen := SaveScreen( 0, 0, MaxRow(), MaxCol() )
  // ← 沒有 cOldColor := SetColor() !
  ...
  SetColor( "W+/B" )      ← 改變色彩
  @ MaxRow(), 0 SAY PadR(...) ← 顯示訊息
  SetColor( cOldColor )    ← cOldColor = NIL → no-op !
  Inkey( 1 )
  RestScreen( 0, 0, ... ) ← 畫面還原但色彩停在 W+/B
```

Harbour 的 SetColor(NIL) 等同 SetColor() (只回傳當前色彩，不改變)，因此 SetColor(cOldColor) 完全沒有恢復色彩的效果，色彩卡在 W+/B。

2.2 影響路徑

MsMenuSel() (主選單導航) 的 DO WHILE 迴圈中直接處理 key 9997/9998/9999：

CASE nKey == 9997

StkTogglePntGoFmt() ← 呼叫後色彩被污染 (W+/B)
 MsMenuSel 繼續執行，選單項目以錯誤色彩顯示

StkCheckSysMenu() 的 idle handler 路徑也受影響 (同樣呼叫此函式) 。

2.3 StkSetPrnMode() — 次要問題

cOldColor 未宣告 LOCAL，成為 PRIVATE MEMVAR，有副作用風險。
 功能上雖可用 (此函式內有正確的賦值和恢復)，但不符合規範。

3. 解決方法

3.1 StkTogglePntGoFmt() — 補上 cOldColor := SetColor()

```
cScreen := SaveScreen( 0, 0, MaxRow(), MaxCol() )
/* 修正：cOldColor 未賦值，補上 SetColor() 以恢復色彩 */
cOldColor := SetColor() ← 新增此行

IF ! hb_FileExists( cIni )
...
SetColor( "W+/B" )
@ MaxRow(), 0 SAY ...
SetColor( cOldColor ) ← 現在能正確恢復
```

3.2 StkSetPrnMode() — 補 LOCAL 宣告

```
LOCAL cContent, nPos, nEnd, cScreen, cMsg
LOCAL cOldColor /* FIX: declare LOCAL to avoid MEMVAR side-effects */
...
cOldColor := SetColor() ← 原本就有賦值，現在改為 LOCAL
```

[OK] 兩處修改均以 Python bytes-only 寫入 (BIG5 中文註解)，BUILD OK

4. 踩坑與陷阱

坑 1：SetColor(NIL) 是 no-op 而非錯誤

Harbour/Clipper 的 SetColor() 若傳入 NIL，等同不傳參數 (只回傳當前色彩)。
 這導致 cOldColor 未賦值的 bug 不會報錯，而是「靜默失效」，
 色彩無聲地被卡在錯誤狀態，難以追蹤。

坑 2：Python replace() 替換第一個出現的 pattern

StkResetPrnSkip 和 StkTogglePntGoFmt 有相同的片段：

```
cScreen := SaveScreen( 0, 0, MaxRow(), MaxCol() )\r\n\r\n IF ! hb_FileExi
```

第一次 fix 時用 data.replace(old, new, 1) 替換到 StkResetPrnSkip (較早出現)，
 而非目標函式 StkTogglePntGoFmt。

解法：先定位函式邊界 (find FUNCTION ... 的 start/end bytes) ，
在 section bytes 內 replace ，確保不會誤中其他函式。

坑 3：替換後 func_end 位移

Fix 2 先寫入 (+78 bytes) ，Fix 1 再寫入時需用 data.find(\r\nFUNCTION ...) 重新定位，
不能用先前快取的 byte offset ，否則定位到錯誤位置。

5. 使用技術與用法

- Python bytes-only 修改 BIG5 PRG (不做 decode/encode ，直接操作 bytes)
- 中文字 BIG5 編碼驗證：word.encode('big5') → 確認 hex bytes 正確
- 精確定位函式邊界：find(b'\r\nFUNCTION StkXxx') + 2 為起點，find(b'\r\nFUNCTION ', start+10) + 2 為終點
- 在 section bytes 內 replace ，避免誤中同名 pattern (不用全域 replace)
- 三次驗證：修改後讀回 bytes ，在正確函式範圍內確認新 pattern 存在

6. 修改檔案清單

檔案：E:\ConvertFW\HBSTK\compat\print_spool.prg

備份：print_spool.prg.bak_setcolor_fix_20260422_2100

- StkTogglePntGoFmt() — 補上 cOldColor := SetColor() (BIG5 中文註解)
- StkSetPrnMode() — 補 LOCAL cOldColor 宣告

7. 結論

[OK] StkTogglePntGoFmt cOldColor 補上 SetColor() ，系統選單操作後色彩正確恢復

主選單 (MsMenuSel) 在用戶點選系統選單項目 (9997/9998/9999) 後，
不再有 W+/B 色彩污染問題。