

CPRNSEL 變數錯誤修正

工作報告

STK 庫存系統 — fw_prnout.prg / FwSelPrnReload

日期：2026-04-22

1. 訴求與現象

在 PDF 預覽視窗中點選印表機選擇按鈕 (selprn)，系統出現 runtime error：

BASE/1003 Variable does not exist: CPRNSEL
Called from FwSelPrnReload(524)

點選後無法正常切換印表機，程式中斷。

2. 根因分析

2.1 cPrnSel 為未宣告變數

[BUG] FwSelPrnReload() 使用 cPrnSel 但未宣告，Harbour 找不到此變數 → BASE/1003

問題程式碼 (line 524)：

```
FUNCTION FwSelPrnReload( cPrnFile, oWebView )  
...  
cPrnNow := IIF( ValType(cPrnSel)="C" .AND. !Empty(cPrnSel), cPrnSel, PrnGetName() )  
// ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  
// cPrnSel 從未宣告，也不是參數 → BASE/1003 Variable does not exist: CPRNSEL
```

FwSelPrnReload 是獨立函式 (FUNCTION)，不是 code block，
無法捕捉 FwPrnPreview 內部的區域變數。
cPrnSel 因此完全不存在於 FwSelPrnReload 的作用域中。

2.2 WebView2 selprn 訊息格式

selprn 觸發路徑：

```
oWebView.bOnBind := {aMsg|  
  IIF( aMsg[1] == "selprn", FwSelPrnReload(cPrnFile, oWebView), ;  
  IIF( aMsg[1] == "pagesetup", FwPageSetup(...), ...  
}  
  
// aMsg 內容只有一個元素：{"selprn"}  
// aMsg[1] = "selprn"，aMsg[2] 不存在
```

JavaScript 端只傳送 'selprn' 字串，不夾帶印表機名稱。
 印表機選擇由 PrinterSetup() 的 Windows 對話框完成，
 選完後可用 PrnGetName() 直接取得當前選定的印表機名稱。

2.3 第一次錯誤修法 (加 aMsg[2] → Bound error)

[BUG] 將 cPrnSel 加為第三參數並傳入 aMsg[2]，但 aMsg 只有 1 個元素 → BASE/1132 Bound error

```
// 錯誤修法：
FUNCTION FwSelPrnReload( cPrnFile, oWebView, cPrnSel ) ← 加了第三參數
...
FwSelPrnReload(cPrnFile, oWebView, aMsg[2]), ← aMsg[2] 不存在！

// 結果：BASE/1132 Bound error: array access
//   aMsg={"selprn"} length:1, 存取 [2]
```

3. 解決方法

3.1 正確修法：直接用 PrnGetName()

由於 PrinterSetup() 會更新系統預設印表機，
 FwSelPrnReload 只需呼叫 PrnGetName() 即可取得使用者選定的印表機，
 完全不需要 cPrnSel 變數。

```
// 修正後 ( line 524 )：
/* 修正: cPrnSel 為未宣告變數, 改以 PrnGetName() 取得選中印表機 */
cPrnNow := PrnGetName()
```

3.2 三處 revert 確認

一共修改三處 (全用 Python bytes-only · BIG5 中文註解)：

Fix 1: FwSelPrnReload 保持 2 個參數 (移除錯誤加入的 cPrnSel)
 FUNCTION FwSelPrnReload(cPrnFile, oWebView)

Fix 2: 呼叫端不傳 aMsg[2] (移除錯誤修法)
 FwSelPrnReload(cPrnFile, oWebView),

Fix 3: line 524 直接用 PrnGetName() (核心修正)
 cPrnNow := PrnGetName()

[OK] 三處驗證全部正確，BUILD OK，印表機切換功能正常

4. 踩坑與陷阱

坑 1：FUNCTION vs code block 作用域差異

Harbour 的 FUNCTION 是獨立作用域，無法存取呼叫端的 LOCAL 變數。
 若需要共用資料，必須透過參數傳遞，或改用 code block ({ |...| ...})。

本案 FwSelPrnReload 是獨立 FUNCTION，
cPrnSel 是 FwPrnPreview 的 LOCAL，兩者完全隔離。

坑 2：selprn WebView2 訊息只有 1 個元素

JavaScript 端 clickSelPrn() 只送出字串 'selprn'，
bOnBind 收到 aMsg = {"selprn"}，長度為 1。
試圖存取 aMsg[2] 立即造成 BASE/1132 Bound error。

解法：不從 WebView2 訊息取得印表機名稱，
改用 PrinterSetup() + PrnGetName() 的系統機制。

坑 3：PrinterSetup() 已處理選擇邏輯

FwSelPrnReload 內部會呼叫 PrinterSetup()，
此函式開啟 Windows 印表機選擇對話框，使用者確認後自動更新系統預設。
PrnGetName() 讀取的就是這個結果，
原本設計的 cPrnSel 根本多餘。

5. 使用技術與用法

- Python bytes-only 修改 BIG5 PRG (不做 decode/encode)
- 精確定位函式邊界後在 section 內 replace，避免誤中其他函式
- PrinterSetup() — 開啟 Windows 印表機對話框並更新系統選擇
- PrnGetName() — 取得當前選定的印表機名稱
- WebView2 bOnBind — aMsg 陣列結構需配合 JavaScript 端實際傳送內容

6. 修改檔案清單

檔案：E:\ConvertFW\PRNASK\source\fw_prnout.prg

備份：fw_prnout.prg.bak_cprnsel_fix_20260422_2130

- FwSelPrnReload() line ~472 — 函式簽名保持 2 個參數 (revert 錯誤的第三參數)
- FwPrnPreview() bOnBind 呼叫端 — 移除錯誤加入的 aMsg[2] 傳遞
- FwSelPrnReload() line 524 — cPrnSel 邏輯改為直接呼叫 PrnGetName()

7. 結論

[OK] FwSelPrnReload 改用 PrnGetName()，印表機切換不再 crash

PDF 預覽視窗中點選印表機選擇按鈕 (selprn) 後，
正確開啟 Windows 印表機對話框，選擇後印表機名稱正確更新，
不再出現 BASE/1003 或 BASE/1132 錯誤。