

鍵盤輔助視窗 滑鼠點選穿透根治

日期：2026-07-23 | 檔案：TLIB.PRG (ShowHotKeys)、mouse_handler.prg (HitTest/MsReadKey)

一、訴求與現象

報價製單表身, 按 F12 或點畫面上方抬頭叫出「鍵盤按鍵一覽表」(PickList), 用滑鼠點選其中一項(如 Alt+F3 倉庫視窗)後, 發生兩個穿透問題: 光棒亂跳——原本停在第 12 筆, 點完跳到滑鼠所在的第 8 筆; 多送一個 ENTER——離開該項目後系統多送一個 ENTER, 若表身正停在「修改」選項就立刻誤入修改模式。

★使用者提供的關鍵線索: 用「點抬頭」進入會多送 ENTER, 用「F12」進入則不會。這個差異最終直接指向真兇。

二、踩坑與陷阱（大量試錯）

錯誤嘗試	為何失敗
在 ShowHotKeys 用 InKey drain 吞掉殘留滑鼠事件	把 K_LBUTTONUP 吃掉→滑鼠按鍵狀態沒「放開」→同位置狂噴 ENTER 洪水, 且被點項目不執行
在 PickList 選取後消耗配對的 K_LBUTTONUP	同上, 吃 UP 破壞按鍵狀態; 且當時搭 hb_keyPut 還有殘留 LastKey 問題
以為 K_LBUTTONUP 是真兇	UP 在 GETSYS(只有 DOWN 的 CASE)/BottomMenu(exit-key 不含 UP)本來就無害, 不是真兇

教訓: 真兇不是 Alt+F3 點擊的放開事件, 而是「觸發鍵盤輔助視窗那個點擊」殘留的按下(DOWN)。診斷靠逐層加 WriteDebugLog(GetApplyKey 每個鍵、BottomMenu、QUOT:698、StockWin)沿事件流回溯, 並要求「一次乾淨重現」避免多次操作疊混。

三、解決方法（兩個獨立根因）

根因 1 — 光棒亂跳:PickList 以滑鼠點選後 LastKey() 停在 K_LBUTTONDOWN;ShowHotKeys 原用 hb_keyPut 把熱鍵排入佇列(尚未執行),控制權先回到表身迴圈,QUOT.PRG:698 的「CASE LastKey()==K_LBUTTONDOWN」讀到這個殘留值,把光棒移到 MRow()(滑鼠列)。

修法:ShowHotKeys 對「有 SetKey 的熱鍵」改成就地 Eval 執行——項目(如倉庫視窗)立即跑,其內部讀鍵會更新 LastKey(),回到表身時不再是 K_LBUTTONDOWN;導覽鍵(PgDn/ESC 無 SetKey)仍走佇列。

```
bHotBlk := SetKey( aKeyCode[nChoice] )
IF bHotBlk != NIL
    Eval( bHotBlk, ProcName(), ProcLine() ) /* ■■■■■,■■■■■ */
ELSE
    hb_keyPut( aKeyCode[nChoice] ) /* ■■■■■■■■ */
ENDIF
```

根因 2 — 多送 ENTER:點抬頭是點在 TopMessage 的 persist 熱區({||ShowHotKeys()});mouse_handler.prg 的 HitTest 執行完 action(Eval)後 RETURN NIL,而 MsReadKey 誤判「沒點到任何區域」→ 把原始 K_LBUTTONDOWN 傳給下層 → 這個殘留按下離開後被表身/GET 誤判成點擊而多送 ENTER。F12

是鍵盤、沒有這個點擊,所以不會——完全對應使用者的線索。

修法:HitTest 在執行 action 時設 lHitAction 旗標;MsReadKey 據此判斷「點擊已被 action 消化」,直接 LOOP 吃掉,不再把 raw K_LBUTTONDOWN 傳下去。

```
/* HitTest ■■ action ■ */ ::lHitAction := .T. ; Eval(aItem[9])
/* MsReadKey IsClick ■■ */
aHit := ::HitTest()
IF ::lHitAction ; LOOP /* ■■■■■■■■,■■ raw DOWN */
ELSEIF aHit != NIL ; hb_keyPut(aHit[4]) ; LOOP
ELSE ; RETURN Int(nKey) ; ENDIF /* ■■■■■■■■■■■■■■■■ */
```

四、使用技術與用法

技術	用途
LastKey() 殘留辨識	看穿「表身迴圈以 LastKey() 判斷點擊」被 PickList 消化的點擊污染
SetKey 熱鍵就地 Eval	讓被點項目先執行、更新 LastKey(),取代排佇列的延後執行

HitTest action 旗標 lHitAction	區分「點到有動作的熱區(已消化)」與「點到空白(該回傳原始點擊)」
逐層 WriteDebugLog 事件流回溯	GetApplyKey/BottomMenu/QUOT:698/StockWin 沿路記錄, 鎖定真兇 DOWN
Python bytes-only BIG5	安全改中文;避開檔內既存 UTF-8 損壞註解(U+FFFD)

操作:報價表身 → F12 或點抬頭叫鍵盤輔助 → 滑鼠點任一項 →
項目正常執行、光棒留在原列、離開後不再多送 ENTER。兩支檔案皆純
Stk32 端;mouse_handler.prg 的 HitTest/MsReadKey
是所有「熱區點擊」共用路徑,此修對點抬頭類觸發全面生效。