

P5141 / P5142 F9 查詢條件 In-Place Refresh

撰寫日期：2026-05-19 專案：HBSTK (Harbour GTWVT Console)

一、訴求與現象

P5141 (國內訂單未出貨明細) 與 P5142 (國內訂單未出貨備品) 進入後會以預設條件查詢，進入瀏覽畫面 (TBrowse) 後，使用者希望按 F9 重新輸入查詢條件 (日期/客戶/料號等)，刷新成新條件的資料，且整個過程不要跳出再回來。

原本的做法是：F9 → LOOKBLOCK 回 .F. → Table RETURN → 外層 loop 重新跑 ReptForm → 重新進入 Table。但實測結果是：

編號	現象
1	輸入新條件後，畫面仍顯示舊查詢結果
2	F9 按下後 AskBox 跳出沒問題，但回到瀏覽畫面後資料沒換
3	有時新舊 Browse 視窗視覺重疊，看起來像有兩個 browse
4	P5141 多次 F9 沒退出 Table，只有按 ESC 中止才退出

二、踩坑與陷阱

坑	症狀	根因
	codeblock 內 @cBeginId 等 by-ref 參數對外層 LOCAL 無效	Harbour codeblock 的 detached local 不支援 @ by-ref。在 block 內呼叫 AskDomOrd1(..., @cBeginId, ...) 雖然能執行，但 AskDomOrd1 改動的值寫不回外層的 cBeginId。
	block 多語句 RETURN 值在某些路徑下不可靠	‘{ nState := X, lF9Refresh := .T., .F. }’ 在 nState=4 (AskBox 完成碼) 時，Table 端的 ‘IF !Eval(LOOKBLOCK) RETURN’ 沒觸發；只有 nState=0 (abort) 才有觸發。推測是多語句 + 寫 detached local 的組合導致回傳值被覆蓋或被 Harbour 評估器吃掉。

坑	症狀	根因
	LastKey() 被 AskBox 改成 1004，不是 K_F9	外層原本靠 'IF LastKey() == K_F9' 判斷是否要 LOOP 重做查詢，但 AskBox 內部有自己的 ReadModal/Keypoll，完成後 LastKey() 是內部完成碼 1004，導致外層誤判 → 進入 'IF lAllData EXIT' → P5141 整個函式退出。
	看到的「舊資料」其實是函式重新被呼叫	因為 讓 P5141 退出，使用者在主選單按 ENTER 再進 P5141，lAllData 重設 .T. → 查詢預設範圍（720天）→ 看起來像是「F9 沒生效」。
	畫面殘留舊 browse	即使流程修對了，退出 Table 再重畫 browse 仍會出現舊 browse 像沒被清掉、新 browse 疊上去的視覺感。

三、解決方法（In-Place Refresh）

放棄「退出 Table → 外層 loop 重做」的路線，改成：不退出 Table，於 LOOKBLOCK 內原地刷新 oBrowse 的資料。

流程：

步驟	動作
1	進入 Table 前，把所有 F9 需要的 LOCAL (cBeginId/cEndId/dates/cReptAlias/cOutputAlias 等) 打包成 aQryRef 陣列
2	LOOKBLOCK 設成 '{ F9RefreshXXX(cAlias, aQryRef, oBrowse)}'，呼叫 helper STATIC FUNCTION
3	helper 內呼叫 AskDomOrdN 拿新條件（用 helper 自己的 LOCAL + @by-ref 是有效的，因為不是 detached local）
4	若使用者 abort (nState=0) → RETURN .T.，保留舊資料留在 Table
5	關閉舊 cReptAlias (dbClearRelation + dbCloseArea)
6	重跑 ReptForm (P5142 含 ExtractParts)，會建立同名 alias（如 DomOrdR1）但內容是新條件查到的資料
7	重綁 dbSetRelation (Customer / Parts / Brand)
8	'oBrowse:refreshAll() + oBrowse:forceStable()' 強制重畫

步驟	動作
9	helper RETURN .T. → Table 繼續執行（不 RETURN）→ 同一個 browse 視窗原地刷新成新資料

****為什麼同名 alias close + reopen 後 oBrowse 仍能正確顯示新資料？****

oBrowse 的 column block 是 ‘{||(cReptAlias)->fldName}’ 形式，引用的是 alias name（字串）。close 舊 alias 後再以同名 reopen，column block evaluate 時找到的是新 alias 的工作區，讀到的就是新資料。refreshAll() 會清快取、重新跑 column block，所以畫面正確更新。

四、使用技術與用法

技術	用法
Carrier Array	把所有需要跨 codeblock 邊界傳遞的 LOCAL 包進一個陣列。陣列是 reference type，block 內讀寫 aRef[n] 都直接影響外層共用的同一個陣列。完成後外層用 aQryRef[n] 取回新值。
STATIC FUNCTION 當 LOOKBLOCK 主體	把多語句 + 控制流程封進一個普通函式，避免 codeblock 多語句 + 寫 detached local 的不可靠性。Block 本體就一行 ‘F9RefreshXXX(...)’，回傳值單純清晰。
oBrowse:refresh All() / forceStable()	refreshAll 清掉 TBrowse 的內部 row 快取並重評估 column block；forceStable 強制走完 stabilization loop 確保畫面繪製完成。兩個搭配是 Harbour TBrowse 重畫的標準組合。
同名 alias replace	ReptForm 內部一律使用固定 alias name（如 DomOrdR1），所以「關掉再開」之後 oBrowse 的 column block 因為以 alias name 為 key，會自動讀到新工作區的資料。這是讓 in-place refresh 可行的關鍵設計。
TABLE.PRG CASE K_F9	原本只有 SEEK 路徑會 Eval LOOKBLOCK。新加一個 ‘CASE nKey = K_F9’ 在前面，不依賴 CHK_SEEK，只要 LOOKBLOCK != NIL 就 Eval；block 回 .T. 留在 Table，回 .F. 才 RETURN 退出。

五、結果

P5141、P5142 兩個畫面 F9 重新查詢功能皆正常：同一個 browse 視窗原地刷新成新條件的資料，視覺上沒有「兩個 browse 疊在一起」的感覺，使用者體驗與原始 CLIPPER 版本一致。