Browse 函式 aMouse/aMsReg 隔離

日期：2026-04-30 專案：HBSTK

一、訴求與現象

從主畫面（如 BUYSELL）透過 SetKey 呼叫 StockWin（ALT+F3）、PartKeyWin（ALT+F4）、FitList（F4）等子視窗時，子視窗的功能提示（WindowMessage 註冊的 aMsReg）與上層呼叫者的 aMouse 混在一起。

* 滑鼠點選子視窗的功能提示時，可能命中外層的 aMouse 區域，觸發錯誤的功能
* 子視窗關閉後，殘留的 aMsReg 項目可能干擾上層的滑鼠操作

二、根因分析

aMouse、aMsReg、nHover 都是 PRIVATE 變數，在呼叫鏈中向下可見。當 BUYSELL 宣告 PRIVATE aMouse 後，FitList 的 ArrayBrowse 看到的 M->aMouse 是 BUYSELL 的，WindowMessage 的新增項目也混入 BUYSELL 的 aMouse/aMsReg。

BrowseKey 的 MsClick 和 HitTest 先檢查 aMouse 再檢查 aMsReg，外層的 aMouse 項目可能先被匹配到。

三、解決方法

在每個 Browse 函式入口宣告 PRIVATE aMsReg:={}, aMouse:={}, nHover:=0，遮蔽上層的同名變數。函式結束時 Harbour 自動釋放內層 PRIVATE，恢復上層的值。

函式	檔案	PRIVATE 位置
ArrayBrowse	ARRAY.PRG	L38
BrrayBrowse	BRRAY.PRG	L36
Table	TABLE.PRG	L42（原有 aMsReg，補 aMouse+nHover）
Browse	BROWSE.PRG	L22

四、踩坑與陷阱

* LOCAL 變數無法遮蔽 PRIVATE — M->aMouse 永遠參照最內層的 PRIVATE，不看 LOCAL

* save/restore 方式（AClone + AEval）比 PRIVATE 遮蔽複雜且容易出錯（EXIT/RETURN 時漏還原）

* PRIVATE 遮蔽是 Harbour 原生機制，函式結束自動恢復，不需手動清理

* TABLE.PRG 原本已有 PRIVATE aMsReg:={}，但缺 aMouse 和 nHover — 只隔離一半

五、使用技術與用法

* PRIVATE 遮蔽：子函式宣告同名 PRIVATE，Harbour 建立新的變數實體，M-> 參照切換到新實體

* 函式結束時，Harbour 自動銷毀內層 PRIVATE，M-> 參照恢復為外層的 PRIVATE

* WindowMessage 透過 MsAddKey 註冊功能提示到 M->aMsReg，BrowseKey 的 HitTest 從 M->aMsReg 讀取

* 隔離後，每個 Browse 層級有自己獨立的 aMsReg/aMouse，互不干擾

六、修改檔案

檔案	行號	修改內容
ARRAY.PRG	L38	PRIVATE aMsReg:={}, aMouse:={}, nHover:=0
BRRAY.PRG	L36	PRIVATE aMsReg:={}, aMouse:={}, nHover:=0
TABLE.PRG	L42	補 aMouse:={}, nHover:=0（原有 aMsReg）
BROWSE.PRG	L22	PRIVATE aMsReg:={}, aMouse:={}, nHover:=0