

STK 滑鼠系統重構工作報告

2026-04-16 ~ 2026-04-17

一、工作概要

將 Clipper 轉 Harbour GTWVT Console Mode 的庫存管理系統 (STK) · 完成滑鼠功能的全面重構。核心成果：建立 MsHandler CLASS · 將所有滑鼠邏輯集中在一個源頭管理。

二、完成項目

2.1 MsHandler CLASS 建立

- 7 個 DATA : nHoverIdx, nSayHoverIdx, lSayRegActive, lInjectQuestion, nInjectTarget, nInjectCount, aRegStack
- 25+ 個 METHOD : BrowseKey, PushReg/PopReg, HitTest, HoverUpdate, HoverCheck, KeyFromLabel, SayReg 系列等
- Singleton 模式：`PUBLIC oMs := MsHandler():New()` 在 main.prg 建立
- 向後相容：所有舊函式名保留為 wrapper · 呼叫端不需改動

2.2 BrowseKey 統一所有 Browse 模組

- 統一處理：hover、右鍵→ESC、雙擊→F10 (area check)、滾輪 (area check)、單擊 (browse 內回傳/browse 外功能提示)
- 套用模組：TABLE, BROWSE, ARRAY, BRRAY, WINFORM, P113, dbEditEx
- 一個源頭：改 CLASS 一次 · 7 個模組全部受益

2.3 MsKeyFromLabel 改用 KY_* defines

- 從硬編碼 `UsrFontGetBig5(0xNNNN)` 改為 `CUSTOMCHAR.ch` 的 `KY\_\*` 定義
- 補齊 F1~F12 全系列 + KY_ALT/KY_ALT2 + KY_CTRL/KY_CTRL2 雙格式支援
- 教訓：永遠不可硬編碼 glyph code · 使用者改 #define 就失效

2.4 PushReg/PopReg 狀態隔離

- 進入修改模式前 `MsPushReg()`：儲存父層 aMsReg/aMouse/hover 狀態
- 離開修改模式後 `MsPopReg()`：還原父層狀態
- 不動 aSayReg：SAY 標頭在修改模式仍需使用
- 解決了 ESC 離開後父層功能提示被擾亂的問題

2.5 dbEditEx 取代 DBEDIT

- 用 TBrowse-based `dbEditEx()` 取代 Harbour 內建 DBEDIT
- DBFWIN.PRG / WINDOW.PRG 改呼叫 dbEditEx
- 使用 `MsBrowseKey(0, oBrowse)` 作為 idle 等待
- 滾輪限 browse 區域內 + 改為 up/down (不是 pageUp/pageDown)

2.6 WindowMessage 升級

- PROCEDURE → FUNCTION (可回傳值)
- 加入 aLocal 陣列 (hover 用的 aMouse)
- 加入 cHintColor + MsAddKey 顏色參數
- 加入 nRow aMsReg 清除 (避免殘留)
- 加入 nWindowRight 邊界截斷 (防止超出邊框)
- 呼叫端改為 `M->aMouse := WindowMessage(...)`

2.7 SayReg 套用 11 個模組

- QUOT, INQUIRY, ENQUIRY, DOMPUR, FORPUR, DOMARR, FORARR, FORCMP, DOMCMP, DOMORD, DOMSCH
- 每個模組的 SAY+GET 標頭自動提取並註冊

2.8 其他修正

- GETSYS QuickBox ELSEIF 限縮：不吞 K_PGDN/K_ESC 等功能提示 key
- GETSYS K_RBUTTONDOWN → hb_keyPut(K_ESC)：右鍵在 ReadModal 正確觸發 ESC + RestScreen
- WINFORM/TABLE 滾輪 GetList 更新：WINDOW_FORM_MODE 滾輪同步右側 form
- idle handler 1006→F10 移除：print_spool.prg 的 StkCheckSysMenu 無條件轉換已移除

三、踩過的坑

3.1 硬編碼 glyph code

- 問題：MsKeyFromLabel 用 `UsrFontGetBig5(0xED58)` 硬編碼，使用者改 `#define KY\_ALT` 就失效
- 表現：只有前 4 個功能提示可點擊
- 解法：全部改用 CUSTOMCHAR.ch 的 KY_* 定義

3.2 GETSYS QuickBox ELSEIF 吞 key

- 問題：ELSEIF 條件太寬，攔截所有 GETINKEYBLOCK != NIL 的 click，K_PGDN/K_ESC 被靜默吞掉
- 表現：修改模式功能提示點擊沒反應
- 解法：ELSEIF 條件限縮為只攔截 K_CTRL_DOWN/UP/ENTER/LEFT

3.3 PRIVATE aMsReg 在 ApplyKey 內的 scope

- 問題：WINFORM 在 ApplyKey 內建 PRIVATE aMsReg，WinFormMessage 更新的是 PRIVATE (即將釋放)，不是父層
- 表現：ESC 離開後父層功能提示被修改模式的 hint 覆蓋
- 解法：PushReg/PopReg 取代手動 aOuterMs save/restore

3.4 PushReg 清空 aSayReg

- 問題：PushReg 儲存+清空 aSayReg，但修改模式的 SAY 標頭需要它
- 表現：修改模式 SAY 標頭點擊失效
- 解法：PushReg/PopReg 不動 aSayReg

3.5 dbEditEx stabilize loop 搶事件

- 問題：stabilize loop 的 `Inkey()` 搶到 1006/K_LBUTTONDOWN，繞過 BrowseKey
- 表現：browse 外事件不受 CLASS area check 控制

- 解法：stabilize loop 過濾所有滑鼠事件 (1001-1015)

3.6 idle handler 無條件 1006→F10

- 問題：print_spool.prg 的 StkCheckSysMenu idle handler 用 hb_keyNext 偷看 buffer，無條件把 1006 轉成 K_F10
- 表現：browse 外雙擊觸發 F10，所有 log 都捕捉不到 (idle 在 InKey 內部執行)
- 偵錯過程：耗時數小時，BrowseKey/SWITCH/stabilize 全部加 log 都看不到，最後搜尋 `hb\_idleAdd` 才找到
- 解法：移除 idle 的 1006 handler (CLASS BrowseKey 已處理)

3.7 LastKey() 差異

- 問題：BrowseKey 的 dblclick 直接 RETURN K_F10，但 LastKey() 是 1006 (不是 K_F10)
- 表現：P113 F9→雙擊選擇後多出 MessageBox
- 解法：改用 hb_keyPut(K_F10) + LOOP，確保下次 InKey 返回 K_F10 且 LastKey() 正確

3.8 nHdrH2 未賦值

- 問題：debug log 在 nHdrH2 賦值前就使用它，`oBrowse:nTop + NIL` → Argument error
- 表現：第一次 K_LBUTTONDOWN 就崩潰，後續所有滑鼠功能失效
- 解法：變數必須在使用前賦值

3.9 K_LBUTTONDOWN 是雙擊的第一擊

- 問題：雙擊 = K_LBUTTONDOWN + K_LDBLCLK，第一擊先到就觸發功能提示
- 表現：雙擊 FORM 區觸發 F10
- 解法：K_LBUTTONDOWN 在 CLASS 內做 area check，browse 外不回傳

3.10 WindowMessage 超出邊框

- 問題：WindowMessage 的 SAY + MsAddKey + aLocal 沒限制在 nWindowRight 內
- 表現：hover 變色超出視窗邊框
- 解法：SAY 截斷 + MsAddKey/aLocal 用 Min(COL()-1, nWindowRight-1) + 截斷文字

四、技術架構

4.1 MsHandler CLASS 結構

```

CLASS MsHandler
  DATA nHoverIdx, nSayHoverIdx, lSayRegActive, ...
  DATA aRegStack    // PushReg/PopReg stack

  METHOD BrowseKey(nTimeout, oBrowse) // 統一 browse InKey
  METHOD PushReg() / PopReg()         // 狀態隔離
  METHOD HitTest() / HoverUpdate()    // aMsReg 系統
  METHOD HoverCheck() / MsClick()     // aMouse 系統
  METHOD SayReg*()                    // SAY 標頭系統
  METHOD KeyFromLabel()               // KY_* 解析
ENDCLASS

```

4.2 BrowseKey 事件處理表

事件 | browse 內 | browse 外

K_MOUSEMOVE | HoverCheck + HoverUpdate | 同

K_RBUTTONDOWN | → K_ESC | → K_ESC

K_LDBLCLK(1006) | hb_keyPut(K_F10) | LOOP (消化)

K_MWFORWARD/BACKWARD | RETURN (caller 處理) | LOOP (消化)

K_LBUTTONDOWN | RETURN (行選擇) | MsClick/MsHitTest (功能提示) 或消化

其他 | RETURN | RETURN

4.3 Hover 配合要求

- Caller 必須宣告：`PRIVATE aMouse := {}, nHover := 0`
- BottomMessage/WindowMessage 必須回傳 aLocal → `M->aMouse := ...`
- 兩個 hover 系統並行：HoverCheck (aMouse) + HoverUpdate (aMsReg)

4.4 修改模式隔離流程

```

MsPushReg()          // 儲存父層 aMsReg/aMouse · 清空 ( 不動 aSayReg )
BottomMessage(修改hints) // 在清空的 aMsReg 上註冊
SayRegEnable()
ReadModal(GetList)
SayRegDisable()
MsPopReg()           // 還原父層
WinFormMessage()     // 重繪父層 hints

```

五、修改檔案清單**檔案 | 修改內容**

mouse_handler.prg | MsHandler CLASS 全面重寫
 main.prg | PUBLIC oMs singleton
 GETSYS.PRG | QuickBox 限縮 + 右鍵→ESC
 WINFORM.PRG | BrowseKey + PushReg/PopReg + 滾輪 + stabilize mouse filter
 TABLE.PRG | BrowseKey + 滾輪 GetList
 BROWSE.PRG | BrowseKey
 ARRAY.PRG | BrowseKey
 BRRAY.PRG | BrowseKey
 P113.PRG | BrowseKey
 DBFWIN.PRG | dbEditEx + aMouse/nHover + callback area check
 WINDOW.PRG | dbEditEx + WindowMessage 升級 + aMouse/nHover
 dbedit_mouse.prg | dbEditEx 新增 (TBrowse-based DBEDIT 替代)
 stk32.hbp | 加入 dbedit_mouse.prg
 print_spool.prg | 移除 idle 1006→F10
 QUOT.PRG | SayReg 註冊
 INQUIRY.PRG | SayReg 註冊
 ENQUIRY.PRG | SayReg 註冊
 DOMPUR.PRG | SayReg 註冊
 FORPUR.PRG | SayReg 註冊
 DOMARR.PRG | SayReg 註冊
 FORARR.PRG | SayReg 註冊
 FORCMP.PRG | SayReg 註冊
 DOMCMP.PRG | SayReg 註冊

DOMORD.PRG | SayReg 註冊

DOMSCH.PRG | SayReg 註冊

六、核心教訓

1. 一個源頭管理：所有滑鼠邏輯集中在 CLASS，不散落各模組
 2. 不可硬編碼：glyph code 用 #define，改定義自動跟著變
 3. idle handler 是隱形殺手：hb_idleAdd 的 handler 在 InKey 內部執行，會偷走/注入 key，所有外部 log 都抓不到
 4. LastKey() 很重要：hb_keyPut + LOOP 確保 LastKey 正確，直接 RETURN 會讓 LastKey 停在原始事件
 5. stabilize loop 會搶事件：InKey(timeout, INKEY_ALL) 在 stabilize 中會搶走滑鼠事件，必須過濾
 6. PRIVATE scope 陷阱：函式內的 PRIVATE 在函式返回前才釋放，WinFormMessage 更新的是 PRIVATE 不是父層
 7. 雙擊 = 兩個事件：K_LBUTTONDOWN 先到，K_LDBLCLK 後到，兩個都要處理
 8. 變數先賦值再使用：未初始化的 LOCAL 是 NIL，算術運算會 Argument error
-

報告完成：2026-04-17

備份：HBSTK_20260417_dbeditex_done