BottomMessage 還原 Alt+H 第二頁功能提示分頁

專案：HBSTK（轉換目標2 / Harbour GTWVT Console） 日期：2026-06-04 17:08

一、訴求與現象

P11 `WINFORM()`（實際定義於 WINFORM.PRG）與 P113 `PARTSFORM()` 兩個主畫面，底部都有一排按鍵功能提示。提示項目多達十幾個，一行 80 欄塞不下，原本設計是用 Alt+H
在「第一頁／第二頁」提示之間切換。轉換為 Harbour 後，按 Alt+H 第二頁的提示完全不出現，等於切換失效。

二、踩坑與陷阱

踩坑 / 陷阱	說明
Alt+H 處理點看似正確卻無效	WINFORM / PARTSFORM 的 Alt+H 只翻轉 `MEMVAR->lHelp` 後重畫底部提示，看起來邏輯沒問題。
Message 函式根本沒讀 lHelp	`WinFormMessage` / `PartsMessage` 只是組提示陣列丟給 `BottomMessage()`，本身完全沒參考 lHelp；真正該分頁的是 BottomMessage。
轉換時整段分頁邏輯被砍掉	HBSTK 版 BottomMessage（TLIB.PRG）留有註解「用一行而已；不再用 lHelp」，原始 `IF MEMVAR->lHelp` 兩頁分支被刪除，只剩單行顯示且 `nCol >= MaxCol()` 就 EXIT，溢出提示既不顯示也不註冊熱區。
原始第二頁分支漏註冊滑鼠熱區	對照 ClassStk 原始碼，page-2 分支只有 @SAY 沒呼叫 `MsAddKey`，若直接照搬會讓第二頁的提示滑鼠點不到。

三、解決方法

在 `BottomMessage()`（HBSTK/source/TLIB.PRG）重建 lHelp 分頁，採「雙掃描」設計：

步驟	作法
1. 算總頁數	第一次掃描依各提示寬度累計，超過 MaxCol() 就換頁、欄位歸零，得出 nMaxPage。
2. 決定目前頁	nPage := IIF(MEMVAR->lHelp, 2, 1)；若 nPage 大於 nMaxPage（沒有第二頁），退回第一頁並復原 lHelp := .F.（沿用原始防呆精神，避免提示列變空白）。
3. 只畫目前頁	第二次掃描用「與步驟 1 完全相同」的分頁規則，只畫出屬於目前頁的項目，且每一頁都正確呼叫 MsAddKey 註冊滑鼠熱區（補上原始第二頁的缺漏）。

四、使用技術與用法

項目	內容
對照法	以 ClassStk/source/TLIB.PRG:1554-1586 原始 Clipper 版為鐵證，確認轉換前確有 lHelp 兩頁分支，據此還原而非憑空猜測。
分頁演算法	換頁判斷統一為 nTestCol > 0 .AND. nTestCol + Len(item) >= MaxCol()，兩次掃描共用同一條件，確保頁碼一致。
BIG5 寫入	新增之中文註解一律以 Python bytes-only 將字串 encode(big5) 後寫入，禁用 Edit/Write 直接寫中文，避免混入 UTF-8 造成畫面亂碼。
驗證 / 建置	kill Stk32 → hbmk2 stk32.hbp -comp=bcc → 啟動實測；進 P11 / P113 畫面按 Alt+H 可在第一頁／第二頁提示間切換，第二頁的滑鼠熱區同步生效。

五、修改檔案

- HBSTK/source/TLIB.PRG — BottomMessage()：新增 4 個 LOCAL（nPage / nCurPage / nTestCol / nMaxPage），ELSE 區塊改為雙掃描分頁。