

密碼輸入 UI 升級 功能報告

模組:RIGHTS.PRГ | 函式:Entry → PassWord → AskPassWord → AskLoginBox | 日期:2026-08-14

一、訴求與現象

系統登入(Entry)在系統參數 FLAG_1 bit0 開啟時會要求輸入使用者名稱與密碼。原本密碼欄位以 HiddenColor() 將前景/背景設為同色來「隱藏」輸入,結果畫面上密碼區只看得到游標移動、沒有任何具體遮蔽字元(星號),使用者無法確認自己到底打了幾個字,體驗不佳。

需求:將密碼輸入 UI 全面升級 —— (1) 星號遮蔽、(2) 登入框美化、(3) 顯示/隱藏密碼切換、(4) 滑鼠與可點提示列。

二、踩坑與陷阱

坑 / 陷阱	說明	影響
Ctrl+H 與退格同碼	原規劃用 Ctrl+H 做顯示/隱藏切換,但 Harbour 中 Ctrl+H == K_BS(退格),兩者同一鍵碼。	會與刪字衝突,改用 F2。
GTWVT 硬體游標不顯示	自繪迴圈中即使 SetCursor(SC_NORMAL/SC_SPECIAL1),在本系統 GTWVT 自繪字型環境下硬體 caret 仍看不到。	使用者反映「輸入區看不到游標」,改為自繪反白游標格。
全形遮蔽字對齊問題	●/* 等全形字寬 2 欄,與 ASCII 密碼字(1 欄)混用會使欄位寬度、游標欄位換算漂移。	改用 ASCII '*' (1 欄),欄位與游標欄 1:1 對齊穩定。
BIG5 中文寫入	PRГ 一律 BIG5,禁止用 Edit/Write tool 直接寫中文(會混入 UTF-8 致亂碼)。	全程以 Python bytes-only、整段 encode('big5') 寫入。

三、解決方法

新增自訂函式 AskLoginBox(@cUserName, @cUserPass, @cCmpyId, lNeedCmpy) 取代原 AskBox 呼叫;AskPassWord 改為只負責判斷欄位數(有無公司代碼)後轉呼叫此函式。核心為一個逐字自繪的 InKey 迴圈:

升級項目	實作方式
星號遮蔽	密碼欄顯示 Replicate('*', Len) 而仍儲存真實字元;達上限不再收字。
美化框	置中排版;外框採系統框線第 4 號 MEMVAR->aFrame[4](@ BOX Left(aFrame[4], 9));標題「系統登入」反白置中;標題/名稱、名稱/密碼、密碼/提示列之間皆留白,版面清爽。
自繪游標	SetCursor(SC_NONE) 關硬體 caret;聚焦欄在插入點畫一格白字紅底(cClrCaret)確保可見。
框內游標移動	維護每欄插入點 aPos[];支援 ←/→/Home/End 移動,於插入點插字、退格刪前字、Del 刪當前字;點欄位可將游標移到點擊處。
顯示/隱藏	F2 切換 lShow;明碼 <-> 星號即時重繪。
滑鼠 + 提示列	底部三段提示各自獨立命中;K_MOUSEMOVE 偵測 hover 段反白(黑/黃), K_LBUTTONDOWN 命中先閃按下色(白/紅)再觸發。

四、使用技術與用法

技術 / API	用途
MEMVAR->aFrame[4]	系統框線第 4 號(chr(21)+chr(23)+...), @ BOX Left(aFrame[4], 9) 畫填滿外框。
SAVE/RESTORE SCREEN	進出登入框保存與還原底層畫面。
DispBegin/DispEnd	整批更新畫面,避免逐格繪製閃爍。
InKey(0) 事件迴圈	統一處理鍵盤(K_ENTER/K_TAB/K_LEFT.../K_F2/K_BS/K_DEL) 與滑鼠(K_MOUSEMOVE/K_LBUTTONDOWN/K_RBUTTONDOWN)。
MRow()/MCol()	取滑鼠儲存格位置,做提示列 hover/click 與欄位聚焦命中判斷。
自繪游標	SetColor 反白單格覆寫插入點,取代不顯示的硬體 caret。
傳參考寫回	cUserName/cUserPass/cCmpyId 皆 @ 傳參考,確認後 Left(PadR(...), N) 依原長度寫回。

相容性:另一呼叫端 APPSETUP.PRG:599

僅傳兩參數(無公司代碼),自動走兩欄版面,已驗證相容;登入流程(PassWord
的 Rights 比對邏輯)完全不變,僅置換輸入 UI。