

P48() 存提款作業 進入速度優化

報告日期：2026-08-19 專案：HBSTK (Harbour GTWVT Console Mode)

一、訴求與現象

進入「存提款作業(P48)」時，主視窗 BROWSE 畫面會「一行一行地刷」，一列一列往下顯示而非一次畫滿；資料列數越多，逐行刷屏的現象越明顯，造成進入時視覺延遲、體感遲鈍。需求是讓進入時的畫面一次繪製完成。

二、根因分析

程式進入 MainWinBody 後的繪製流程存在「對稱性缺口」：

對象	進迴圈前處理	結果
子視窗 oSubWin	goBottom() + forceStable() (第98-99行)	一次畫滿，正常
主視窗 oMainWin	無 forceStable，僅靠主迴圈 stabilize() 逐列繪製	逐行刷屏

主迴圈內的穩定化邏輯為：

```
DO WHILE nKey = 0 .AND. !oMainWin:stable
```

```
    oMainWin:stabilize() // 每次只畫一列
```

```
    nKey := InKey(0.01, INKEY_ALL) // 每畫一列停 10ms
```

TBrowse:stabilize() 每次呼叫只繪製一列，加上中間夾了 InKey(0.01)

每列等待 10 毫秒，於是進入時整個 browse 就變成逐列、帶延遲地繪製。

三、踩坑與陷阱

項目	說明
非新錯誤	子視窗早已在第99行使用 forceStable()，只是主視窗漏了；屬既有對稱性缺口，非本次改動引入。
勿改動迴圈邏輯	InKey(0.01) 輪詢在執行期(按鍵後)仍有用途，不可移除；只需在「進入時」補一次強制穩定即可。
中文註解編碼	PRG 為 BIG5，禁止用 Edit tool 直接寫中文，改用 Python bytes-only 以 BIG5 寫入，避免整頁亂碼。

四、解決方法

在進入主迴圈前（第106行 `oMainWin:autoLite := .F.` 之後），比照子視窗補上一行強制穩定：

位置	新增程式碼
MainWinBody 第107行	<code>oMainWin:forceStable()</code> // [Claude] 進入時一次畫滿主視窗, 避免逐列刷屏

`forceStable()` 會一次把整個 `browse` 穩定並繪製完成；之後主迴圈因 `oMainWin:stable` 已為真，內層 `stabilize` 迴圈不再逐列重畫，進入時即一次顯示完整畫面。此手法與子視窗完全一致，不改任何流程邏輯，風險極低。

五、使用技術與用法

技術	用法
<code>TBrowse:forceStable()</code>	強制一次完成 <code>browse</code> 穩定與繪製，取代逐列 <code>stabilize()</code> ；適用於「進入 / 重整後」需要立即顯示完整畫面的時機。
<code>stabilize()</code> vs <code>forceStable()</code>	<code>stabilize()</code> 每呼叫一次只畫一列(可被按鍵中斷，適合互動)； <code>forceStable()</code> 一次畫滿(適合初始顯示)。
Python bytes-only 改 BIG5	以 <code>open(wb)</code> 讀出 bytes、 <code>str.encode("big5")</code> 產生中文註解、 <code>bytes.replace</code> 定位插入，全程不做 <code>decode</code> ，避免 UTF-8 混入。

六、修改檔案

檔案	變更
HBSTK/source/P48.PRG	MainWinBody 第107行新增 <code>oMainWin:forceStable()</code> (含 BIG5 中文註解)
建置結果	重建 <code>Stk32.exe</code> : BUILD OK，已啟動測試，逐行刷屏消除