MenuBar 閃爍修復技術報告

日期：2026-04-25　專案：HBSTK (GTWVT Console Mode)

一、訴求與現象

系統選單（PullDownMenu）在子選單之間切換時，最上方的 MenuBar 會明顯閃爍。每次按 LEFT/RIGHT 從一個子選單移到另一個子選單時，MenuBar 整個被重繪一次，肉眼清楚可見閃動。

使用者回饋：「每次次選單移動，我在螢幕上看得很清楚，MENUBAR 整個重畫然後閃了閃」

在較慢的電腦上更為嚴重，同時選單框線也會出現殘影。

二、踩坑與陷隱 — 嘗試過的方法

在找到根因之前，嘗試了多種 Windows API 和 Harbour 機制，全部無法解決 MenuBar 閃爍：

方法	原理	結果
WM_SETREDRAW	停止/恢復視窗重繪	無效
LockWindowUpdate	鎖定視窗，阻止所有繪圖	無效
DispBegin / DispEnd	Harbour GT 層級批次暫存	無效
Stuff() buffer	先組好完整字串再一次 SAY	略有改善仍閃
+= 串接 buffer	用 += 取代 Stuff 避免位置問題	略有改善仍閃

問題不在 API。邏輯問題改死 API、用盡天下的 API 也沒用。

三、解決方法 — 流程追蹤找到根因

3.1 逐行流程追蹤法

停止嘗試新的 API，改用「逐行流程追蹤」，在每一行旁邊標註凍結/解凍狀態。缺口立刻浮現：

3.2 修正前的流程（有 BUG）

```
[sub-menu returns]
line 174: FreezeWindow(.T.)     <-- freeze
line 175: RestScreen(...)       <-- restore (MenuBar gone)
line 177: FreezeWindow(.F.)     <-- *BUG: UNFREEZE!
                                    user sees 'no MenuBar'
line 199: Status()              <-- unfrozen, repaint visible
         ... loop back to top ...
line 64:  SaveScreen(...)
line 65:  FreezeWindow(.T.)     <-- re-freeze (too late!)
line 68-100: redraw MenuBar     <-- frozen, invisible
```

```
line 156: FreezeWindow(.F.)     <-- unfreeze, MenuBar appears
```

根因：line 177 解凍 → line 65 重新凍結之間有「裸奔」區間。即使只有幾毫秒，RestScreen 恢復的「空白 MenuBar」畫面已經被顯示出來了 — 這就是閃爍的來源。

3.3 修正後的流程

```
[sub-menu returns]
line 174: FreezeWindow(.T.)     <-- freeze
line 175: RestScreen(...)       <-- restore (frozen, invisible)
                                <-- NO unfreeze, stay frozen
line 199: Status()              <-- frozen, invisible
         ... loop back to top ...
line 65:  FreezeWindow(.T.)     <-- re-lock same window (harmless)
line 68-100: redraw MenuBar     <-- frozen, invisible
line 156: FreezeWindow(.F.)     <-- unfreeze HERE, show complete frame
```

結果：閃爍完全消失，一閃都沒有。

四、使用技術與用法

4.1 FreezeWindow 實作

FreezeWindow() 封裝在 TLIB.PRG #pragma BEGINDUMP 中，使用 Windows API LockWindowUpdate：

```
HB_FUNC( FREEZEWINDOW )
{
   if( hb_parl(1) )
      LockWindowUpdate( GetActiveWindow() );
   else
      LockWindowUpdate( NULL );
}
```

4.2 關鍵特性

- LockWindowUpdate 同視窗重複呼叫是無害的（重新鎖定同一視窗）
- LockWindowUpdate(NULL) 會立即解鎖 — 放錯位置就是閃爍的來源
- 系統同一時間只能鎖定一個視窗
- SaveScreen / RestScreen 操作的是 Harbour GT buffer，不受 LockWindowUpdate 影響

4.3 修改的檔案

- MENU.PRG：移除 3 行 — 刪除 debug sleep、刪除兩處 RestScreen 後的 FreezeWindow(.F.)
- 備份：MENU.PRG.bak_PERFECT_freezecont_20260425（修正後完美版本，不刪除）

五、方法論總結 — 逐行流程追蹤法

當遇到「機制應該生效卻沒效果」的問題時：

1. 停止嘗試新的 API 或 workaround
2. 列出程式碼的完整執行順序（含跨函式、迴圈回頂端、遞迴返回）
3. 在每一行旁邊標註關鍵狀態（凍結/解凍、可見/不可見、buffer 狀態）
4. 用 ★ 標記狀態突變的位置
5. 邏輯缺口自然浮現，不需要猜

問題在邏輯，不在 API。邏輯問題改死 API、用盡天下的 API 也沒用。
先追流程、標狀態、找缺口 — 這才是正確的除錯方向。