

GTWVT 效能提升計畫

日期：2026-04-28 專案：HBSTK

一、訴求與現象

視窗 Resize 後進入 WINFORM 畫面刷新明顯變慢，肉眼可見逐行刷新。
經實測確認：ScanAndDrawUda 呼叫次數 resize 前後相同（非次數暴增），每次 WM_PAINT 從 1ms 變為 4-5ms（BitBlt capture 耗時暴增 5 倍）。

二、根因分析

- * BitBlt capture (screen to buffer) 在 resize 後耗時從 156ms 暴增至 831ms
- * PaintText (GTWVT) 直接畫到螢幕 DC，每行 ExtTextOut 肉眼可見
- * ScanAndDrawUda 每次掃描全部 2000 cells (25x80)，即使 dirty 只有 1-2 行
- * 每次 WM_PAINT 都 CreatedDC/CreateDIBSection/DeleteDC/DeleteObject (per glyph)
- * 生產環境殘留 debug I/O：cursor_debug.log fopen、RestScreen WriteDebugLogC

三、usrfont_c.c 拆檔

原始 3801 行的單一檔案拆成 8 個子檔案（#include 串接，編譯行為不變）：

檔案	行數	內容
usrfont_int.h	92	共享 header：常數、型別、外部宣告
usrfont_c.c	727	主檔：靜態變數 + codepage hooks + #include
usrfont_cac he.c	274	ScaleGlyph + CaptureGlyph + EnsureAlpha + DrawUdaAlphaBlend
usrfont_ren der.c	361	SaveUmap + LoadUmap + DrawOneGlyph + DrawDynUnicode
usrfont_asc font.c	685	ASCFONT 全部渲染和快取
usrfont_sca n.c	431	ScanAndDrawUda 主渲染函式
usrfont_wnd proc.c	182	UsrFontWndProc 訊息處理 hook
usrfont_api .c	1127	所有 HB_FUNC 匯出函式

四、完成的優化（17 項）

編號	優化	效果
A	dirty-range BitBlt	capture/writeback 只處理 dirty 行，CAP 831ms to 142ms
B	DispBegin/DispEnd + WinFormShow	WM_PAINT 批次化，FORM 區域一次輸出
C	PaintText double buffer	快取 memory DC，ExtTextOut 不再逐行可見
D	Pre-scan 無 UDA 跳過	純 ASCII 畫面零 capture/scan/writeback 開銷
E1	DrawUdaAlphaBlend DIBSection 快取	每 glyph 省 4 次 GDI 建/銷
E1b	DrawAscFontFromCache DIBSection 快取	同 E1，用於 ASCFONT
E3	scan loop 只掃 dirty 行	減少 95% hb_gtGetChar 呼叫
E4	cursor_debug.log 移除	消除 WM_PAINT 內 disk I/O
E5	RestScreen WriteDebugLogC 移除	消除 hot path disk I/O
E6	WriteDebugLogC 路徑快取	每次省 GetModuleFileName syscall
E8	移除冗餘 CreateRectRgn	DrawDynUnicode/DrawOneGlyph 省 GDI 物件
E9	CreateSolidBrush 快取	最大化/sizing 不再每次建/銷 brush
E10	usrfont_wcharLen CJK 早期返回	避免 785 筆線性掃描
E11	InvalidateRect 縮小到 caret 行	游標變更不再全畫面重繪
E12	SetTextAlign 設一次	每次 WM_PAINT 只呼叫一次
E13	LoadUsrFontUmap 三重重複移除	消除 dead code
E14	SaveUsrFontUmap 記憶體比對	取代每次寫入前讀檔
E15	USRFONT_DEBUGMAP 停用	啟動時不再寫 log

五、放棄的優化

E7 DestroyCaret/CreateCaret 快取：INSERT mode

左移游標跑到上方。原因：HideCaret/BeginPaint/EndPaint/ShowCaret 的交互破壞 caret 內部狀態，跳過 CreateCaret 後 caret 位置不同步。

六、待完成

完整 double buffer：PaintText + ScanAndDrawUda 共用同一個 off-screen buffer，一次 BitBlt 輸出。目前問題：buffer 未初始化基底內容，PaintText 只畫 dirty 區域導致其餘黑色。修正方向：在 PaintText 前先複製螢幕內容到 buffer。

七、踩坑紀錄

- * BIG5 bytes 含 0x7D ()：BCC32 C89 誤判括號匹配，拆檔後小檔案容易追蹤
- * 移除 cursor_debug 時多留 { ：一行行看 443 行找到
- * FiveWin ON CHANGE { || } 雙層包裝：預處理器展開成 {|Self| { || } }，內層不執行
- * FiveWin DllCall(CB_SETCURSEL) 導致 dialog 關閉：改用 oCboPaper:Select()
- * E7 caret 快取失敗：BeginPaint/EndPaint 干擾 caret hide counter

* 完整 double buffer 畫面黑色：buffer 需基底內容，PaintText 只畫 dirty rect

八、實測數據

指標	修改前 (resize 後)	修改後 (resize 後)
WM_PAINT avg	4-5ms	1-3ms
CAP (BitBlt capture)	486-831ms/2s	0-280ms/2s
UDA 總耗時	267-1155ms/2s	大幅降低
進入 WINFORM 體感	逐行刷新可見	一次刷出，與 resize 前一致