

GTWVT 引擎：造字學習根治與渲染效能

日期 2026-06-06 | 範圍：usrfont_c.c / usrfont_wndproc.c | 非BIG5 Unicode
字輸入 + WM_PAINT 最佳化

一、訴求與現象

1. 注音輸入「BIG5 沒有、但 Unicode 有」的字（如 峯 U+5CEF、鋒 U+93E0）完全進不了 STK32；但同樣不在 BIG5 的 堃 U+5803、喆 U+5586 卻可以——現象很矛盾。

2. 客戶端在 CJK

密集畫面（整片造字/全形字）捲動、打字時，螢幕重繪偏慢。

二、踩坑與陷阱

UMAP 自我學習池：

線索	發現
三字編碼	Harbour BIG5 表對 峯/堃/喆 全回 0x3F(?)，代表都走同一條「?→DynAllocSlot」學習路徑，差別不在編碼。
UMAP 記錄	堃/喆 早已配到 slot 784/783（去重命中既有）才「能用」；峯 整個 UMAP 找不到。
診斷 log	峯：reach DynAllocSlot → slot=-1（配置失敗）；堃：slot=784。證明峯走到了學習、卻配不到空位。
真因	USRFONT.TTF 把整批華康 EUDC (E000 - EE1A 共 3611 字) 搬進來；wcharGet / wcharPut case-1 每存取 FA - FE 任一格，就用 CP950 反查填佔位並 SaveDynMap 持久化。只要開過一次「造字對照表」（掃 E000 - E310），699 格動態池全被佔滿 → DynAllocSlot 回 -1。
鐵律	BIG5 FA40 - FEFE 共 785 碼是硬上限：「搬整批華康」與「留動態學習池」在同一塊碼空間互斥。

畫面渲染：

線索	發現
標準路徑	GTWVT 標準流程 Redraw=InvalidateRect（只排程）、Refresh=idle 才 pump，本來就非阻塞+合批，不是瓶頸。

線索	發現
真凶	usrfont 的 WM_PAINT 殘影偵測：只要任一 UDA cell 內容變動，就把髒區撐成全螢幕(0, 0, 32767, 32767) → ScanAndDrawUda 掃全 25 列×80 欄、重 alpha-blend 整片 CJK，即使只變一格。
非重點	glyph 渲染本身有快取(EnsureUfntAlpha 只算一次)；浪費在「重 blend + 重 blit 沒變的 cell」。

三、解決方法

UMAP（分治標+治本，已壓力測試通過）：

階段	做法
治標(離線清池)	先 kill Stk32（否則退出時用記憶體滿版 map 覆寫 UMAP），Python 清掉指定 FA - FE 範圍 slot（FA44 - FCA4 共 202 格），重啟。峯/縫 立刻回來。
治本 A	wcharGet 與 wcharPut case-1 兩處 SaveDynMap 移除——華康 EUDC 佔位只在記憶體供當下渲染、不再寫進 UMAP（開造字表也不會永久污染）。
治本 B	DynAllocSlot 加第二輪「池滿時回收 PUA 佔位(E000 - F8FF) slot」給真實學習字；真實字(<E000) 永遠受保護不被搶。
驗證	先開造字表再打 峯/縫 照樣進、重開 Stk32 不被污染。

渲染效能：把「任一變動就全螢幕」改成「只 union 真正變動的 cell 所在列」，ScanAndDrawUda 只重畫變動列。打字/狀態列省約 25 倍；捲動全頁時 union 自然涵蓋全部=無回歸；保險機制：變動超過 1/4 螢幕時仍退回全螢幕。

四、使用技術與用法

項目	內容
診斷技術	C 層 WriteDebugLogC 在 wcharPut/WndProc 插 log；小 Harbour 程式呼叫 hb_cdpFind("BIG5")->wcharPut 驗證；Python dump USRFONT.UMAP（785×HB_WCHAR 純 raw、小端）分類 PUA佔位 vs 真實學習字。
修改檔案	c_code\usrfont_c.c（DynAllocSlot 回收 + 兩處不持久化）、c_code\usrfont_wndproc.c（WM_PAINT dirty-row union）。

項目	內容
編譯	stkgt_lib\build_StkGTWVT.bat : gtwvt.c -O2 + usrfont 經 hbm2 -hbcmp(無優化/無 unicode) → tlib 合併 StkGTWVT.lib → 安裝成 Harbour gtwvt.lib → 重建 Stk32。
BIG5 編碼	所有 C/PRG 改動一律 Python bytes-only 操作，中文註解以 .encode("big5") 寫入，禁止 Edit/Write 直接寫中文。