

StkGTWVT 中文雙位元 GTWVT 引擎

使用手冊 | Harbour Console (GTWVT) + 華康造字 + UDA + ASCFONT | 2026-06-04

一、這是什麼

StkGTWVT 是一顆把 GTWVT 終端驅動 (Harbour 用 GDI 視窗模擬 DOS 文字主控台) 與 usrfont 造字引擎 (中文 BIG5 雙位元、華康造字、UDA 動態造字、ASCFONT 點陣 ASCII) 合併在一起的單一靜態函式庫 StkGTWVT.lib。它讓 Clipper/Harbour 程式在 Windows 視窗中以純文字介面 (80x25、16 色) 執行，同時正確顯示中文、自訂造字字、框線，並支援滑鼠。過去 gtwvt 與 usrfont 分屬兩套 build、整合環境複雜。本引擎把它們封成一顆 lib + 一支一鍵 build，目的：未來開新專案能「拿了就用」這顆中文雙位元引擎，不必重建複雜環境。

二、架構總覽

引擎由兩個 C 物件組成，靠 7 個 extern 全域變數橋接，耦合面極窄：

組成	角色	編譯需求
gtwvt.obj (gtwvt.c)	GTWVT 終端驅動：建立視窗、cell 繪製框架、決定何時重畫、滑鼠/鍵盤訊息	-O2 + UNICODE
usrfont.obj (usrfont_c.c)	造字引擎：載入 USRFONT.24M、subclass 視窗、ScanAndDrawUda 把造字/UDA 點陣畫進畫面、BIG5 解碼、IME	無優化 + 無 UNICODE

橋接機制：gtwvt 把 cell 幾何與繪圖 buffer 透過 7 個全域匯出給 usrfont (g_nCellW / g_nCellH / g_nMarginLeft / g_nMarginTop / g_nFontLfWidth / g_bHookPaint / g_hPaintBuf)；usrfont 在自己 subclass 的 WndProc 裡用 ScanAndDrawUda 讀這些值、把造字點陣畫到同一張 bitmap，再一次 BitBlt。gtwvt 不直接呼叫 usrfont 函式——所以兩者能各用各的編譯旗標、合進一顆 lib。

三、在新專案中整合（4 步）

步驟 1：連結引擎 lib 把 StkGTWVT.lib 安裝成 Harbour 的 gtwvt.lib，hbp 沿用 -gtwvt：

```
copy StkGTWVT.lib E:\Harbour32\lib\win\bcc\gtwvt.lib  
(stk32.hbp 內保留一行：-gtwvt)
```

步驟 2：宣告 GT 請求 在 main.prg 最上方：

```
REQUEST HB_GT_WVT
```

步驟 3：啟動時初始化造字引擎 在進入主程式前呼叫（會載入字型並 hook GT/視窗）：

```
UsrFontInit( "USRFONT.24M" ) 載入 CJK 造字字型 + 掛勾  
UsrFont_SetIcon( 1 ) 選配：視窗 icon  
UsrFont_SetWinStyle( .T. ) 選配：[X] 關閉鈕  
UsrFont_SetCursorHeight( nNormal, nIns ) 選配：游標高度
```

步驟 4：附上資料檔 把下表的 data 檔放在 EXE 同目錄。完成。

四、需要的資料檔（放 EXE 同目錄）

檔案	用途	必要性
USRFONT.24M	中文 BIG5 造字字型（24 點陣，主要 CJK glyph 來源）	必要
ASCFONT.UMAP	ASCII 點陣字對照表（byte→Unicode 映射）	必要
ASCFONT.15	ASCII 點陣字 bitmap cache	建議
*.BCACHE	各解析度的 glyph 點陣快取（首次執行可自動產生）	選配（加速）

五、公開 API

(A) GTWVT 驅動層（C 端 HB_FUNC，PRG 直接呼叫）

函式	用途
GTWVT_SETFONTNAME_W(cWName)	以 WCHAR 設定字型名（繞過 BIG5 無法表示的字型名）
GTWVT_SETCELLWIDTH(nW)	DOSBox-X 風格獨立設定 cell 寬度，不扭曲字型
GTWVT_SETEXTWIDESYMBOL(lOn)	把額外符號視為全形（佔 2 cell）
hb_gtInfo(210, nPx) / (211, nPx)	設定一般 / INSERT 游標像素高度

(B) usrfont 造字引擎層（PRG 封裝函式，建議用這層）

函式	用途
UsrFontInit(cFile)	載入 .24M 造字字型，hook GT/codepage/視窗。回傳 .T.=成功
UsrFont_SetIcon(nResID)	設定視窗 icon（資源 ID）
UsrFont_SetWinStyle(lCloseBtn)	開關視窗 [X] 關閉鈕
UsrFont_SetCursorHeight(nNorm, nIns)	設定一般/INSERT 游標高度
UsrFontGetBig5(nUnicode)	Unicode → BIG5 查詢

函式	用途
UsrFontRefresh() / ResetCache()	重新整理 / 清快取（換字型或 cell 尺寸後）

六、要改引擎時：一鍵重建

改了 gtwvt.c 或 usrfont_c.c 後，只要跑一支 bat 就全自動完成：

```
E:\ConvertFW\HBSTK\stkgt_lib\build_StkGTWVT.bat
```

它依序做：[0] kill Stk32/PrnAsk → [1] 編 gtwvt.c(-02) → [2] 編 usrfont(hbmk2 -hbcmp) → [3] tlib 合併 → [4] 裝成 gtwvt.lib → [5] rebuild Stk32 → [6] 啟動。正確的編譯旗標已全部寫死在 bat 內並附註解，不需手動處理。

七、三個致命雷（絕對不要踩）

這三個都是「編得過，但執行期才壞」——編譯成功不代表正確，務必用實機畫面驗證。

雷	症狀	正確做法
usrfont 用 -02 編	框線完全錯亂 + HB_BITOR Argument error 崩潰 (usrfont 有對優化敏感的 UB)	usrfont 一律無優化 (hbp 的 -optim-; bat 用 hbmk2 -hbcmp)
usrfont 加 -DUNICODE	視窗變形、當機 (TCHAR/Windows API 變 W 版，與原本 ANSI 不符)	usrfont 不要 UNICODE；用 hbmk2 -hbcmp 自動正確
GT 機制改 -1XXX+REQUEST	視窗變形（漏掉 -gtwvt 才會引用的 HB_GT_WVT_DEFAULT = 設預設GT+初始視窗）	沿用 -gtwvt，把合併 lib 裝成 gtwvt.lib

取得正確 obj 的鐵則：usrfont.obj 必須用 hbmk2 -hbcmp 取（原班工具鏈的真貨）；不要手寫 bcc 旗標——連「逐項對齊的精確鏡像」都會差 160 bytes 而崩潰。

八、故障排除

現象	可能原因 / 處理
框線錯亂 / HB_BITOR 崩潰	usrfont 被用 -02 編了 → 改回無優化 (hbmk2 -hbcmp)

現象	可能原因 / 處理
視窗變形 / 當機	usrfont 帶了 -DUNICODE，或 GT 改用 REQUEST 漏了 HB_GT_WVT_DEFAULT → 沿用 -gtwvt
中文/造字顯示為空白或方塊	USRFONT.24M / ASCFONT.UMAP 沒放在 EXE 目錄，或 UsrFontInit 未呼叫/回傳 .F.
編譯報「1 errors」卻無 Error 行	bcc32.cfg 的 -w! 把 SDK 標頭 warning 升 error (-O1/-Od 才會) → 用無優化或 -O2
改了 C 碼卻沒生效	忘了跑 build_StkGTWVT.bat (lib 沒重編沒重裝)

附錄、引擎包檔案清單

路徑	內容
stkgt_lib\StkGTWVT.lib	合併引擎庫（裝成 gtwvt.lib 使用）
stkgt_lib\build_StkGTWVT.bat	一鍵重建腳本
stkgt_lib\gtwvt.obj / usrfont_hbmk.obj	中間物件（重建時自動產生）
build\ufonly.hbp	hbmk2 -hbcmp 用：只編 usrfont_c.c 取真貨 obj
gtwvt_patch\gtwvt.c (+.h)	GTWVT 驅動原始碼（要改驅動改這）
c_code\usrfont_c.c (+模組 .c)	造字引擎原始碼（傘狀含 cache/render/ascfont/scan/wndproc/api）
compat\usrfont.prg	PRG 封裝層（UsrFontInit 等公開函式）