

# 造字工作流 + USRFONT.TTF 預覽

STK 庫存系統 客戶端造字 + 開發端字型檢視

|      |                         |
|------|-------------------------|
| 日期   | 2026-05-15              |
| 版本   | 1.0                     |
| 對象   | 客戶造字者 + 開發者 + 系統維運      |
| 前置   | STK 已啟用 FontDirect=4 模式 |
| 核心觀念 | 90% 造字不需動 PRG，純資料層操作    |

## 目錄

| 章節 | 內容                         |
|----|----------------------------|
| 1  | 為什麼要造字                     |
| 2  | 造字的兩種情境                    |
| 3  | 情境 A：客戶資料造字（不需改 PRG）       |
| 4  | 情境 B：系統內建符號（要改 PRG）        |
| 5  | USRFONT.TTF 預覽功能（Ctrl+F12） |
| 6  | FontForge 造字操作指南           |
| 7  | PUA codepoint 配置慣例         |
| 8  | 常見問題 Q&A                   |

## 1. 為什麼要造字

STK 系統內常見的造字需求：

| 類型       | 範例                       | 頻率          |
|----------|--------------------------|-------------|
| 客戶姓名罕用字  | 字「堃」、「煜」、「琚」等不在 Big5 的字  | 高           |
| 公司名特殊字   | 台塑/南亞等舊系統造字              | 中           |
| 產品名稱     | 特殊規格符號                   | 低           |
| 系統介面按鍵圖示 | [Fns] [Del] [PgDn] 等鍵盤圖示 | 低（系統內建一次決定） |

### 傳統做法（倚天時代）

用 ETEN 造字程式畫 24x24 bitmap → 存進 USRFONT.24M → DBF 內存對應 Big5 UDA 碼

### 新做法（GTWVT + USRFONT.TTF）

用 FontForge / FontCreator 畫向量 glyph → 存進 USRFONT.TTF 的 PUA 區 → DBF 仍存 Big5 UDA 碼，自動透過 USRFONT 系統渲染

## 2. 造字的兩種情境

| 項目                        | 情境 A：客戶資料用字                     | 情境 B：系統內建符號                |
|---------------------------|---------------------------------|----------------------------|
| 誰造字                       | 客戶側造字者（IT 人員）                   | 開發團隊                       |
| 頻率                        | 頻繁（隨時可能新增）                      | 稀少（系統設計時一次決定）              |
| 例子                        | 客戶名字「堃」、地名罕用字                   | 鍵盤圖示 [Ins] [Del] [PgDn]    |
| 工具                        | FontForge（或 STK 內建未來 FontForge） | FontForge + Code editor    |
| 動 USRFONT.TTF             | 要（加新 glyph）                     | 要（加新 glyph）                |
| 動 USRFONT.UMAP            | 不需要（runtime 自動成長）               | 不需要（同 A）                   |
| 動 PRG / CustomCharacter.h | 不需要！                            | 需要（加 KY_NEWKEY 等 macro）    |
| 重新編譯 Stk32                | 不需要！                            | 需要                         |
| 重新部署                      | 只要部署 USRFONT.TTF                | 部署 Stk32.exe + USRFONT.TTF |
| 完成時間                      | ~ 30 秒                          | ~ 30 分鐘（含編譯測試）             |

### 關鍵領悟

- 90% 造字屬情境 A — 純資料層，完全不碰 PRG
- 情境 B 是系統擴充，由開發者偶爾做
- 不要混淆兩者 — 客戶造字不該要工程師參與

### 3. 情境 A：客戶資料造字（不需改 PRG）

#### 3.1 操作步驟

| # | 操作                                   | 工具        |
|---|--------------------------------------|-----------|
| 1 | 確認需要的字（如「堃」U+5803 或全新字 U+EE50）       | 客戶需求單     |
| 2 | 開 FontForge，開啟 W:\\S002\\USRFONT.TTF | FontForge |
| 3 | 前往指定 Unicode 位置（PUA 或標準 CJK）         | FontForge |
| 4 | 畫新 glyph（複製貼上 / 手繪向量）                | FontForge |
| 5 | 存檔                                   | FontForge |
| 6 | 複製新 TTF 到 W:\\S002\\USRFONT.TTF（覆蓋）  | Explorer  |
| 7 | 通知客戶重啟 Stk32                         | 電話/Mail   |
| 8 | 客戶在 Stk32 內用 Windows 內碼輸入法 'UEE50 輸入 | 客戶 Stk32  |
| 9 | 畫面立即顯示新字                             | 自動        |

#### 3.2 底層自動運作

■■ IME ■■ "UEE50"

↓

Windows IME → WM\_IME\_CHAR(wParam=U+EE50)

↓

UsrFontWndProc ■■ → DynAllocSlot(0xEE50) → ■ slot

↓

■■ wParam ■ PUA E000+slot

↓

GTWWT ■ PUA → CP950 → Big5 FA-FE bytes → ■■ buffer

↓

ScanAndDrawUda → CaptureGlyphAlpha(U+EE50, USRFONT-STK)

↓

USRFONT.TTF ■ glyph at U+EE50 → ■■ ✓

↓

■■■ Enter → DBF ■ Big5 FA-FE bytes

↓

■■■■ → SAY oRecord:Name → ■■■■ → ■■ ✓

### 3.3 對應更新（不必管，自動完成）

| 元件           | 更新方式                                   |
|--------------|----------------------------------------|
| USRFONT.TTF  | 造字者手動加 glyph                           |
| USRFONT.UMAP | <b>runtime DynAllocSlot 自動填入</b>，跨啟動保留 |
| DBF          | 客戶輸入時自動以 Big5 UDA 儲存                   |
| PRG 程式碼      | <b>完全不動</b>                            |

## 4. 情境 B：系統內建符號（要改 PRG）

此情境只發生在「新增系統級鍵盤圖示」等場合，例：要加 [CapsLk] 按鍵圖示。

### 4.1 操作步驟

| # | 操作                                                                          | 檔案                   |
|---|-----------------------------------------------------------------------------|----------------------|
| 1 | 決定 Unicode codepoint (如 U+ED70 for CapsLk)                                  | -                    |
| 2 | FontForge 加 glyph 到 USRFONT.TTF 的 U+ED70                                    | build/USRFONT.TTF    |
| 3 | CustomChar.ch 加 #define KY_CAPSLK = UsrFontGetBig5(0xED70) to CustomChar.ch | source/CustomChar.ch |
| 4 | PRG 程式碼引用 KY_CAPSLK 顯示                                                      | source/*.PRG         |
| 5 | 重新編譯 Stk32                                                                  | _rebuild_claude.bat  |
| 6 | 部署 Stk32.exe + USRFONT.TTF                                                  | deploy_stk32.bat     |
| 7 | 客戶端執行 → 自動顯示                                                                | -                    |

### 4.2 涉及程式碼變更

```
// CustomChar.ch ■■■■  
  
#define KY_CAPSLK ( UsrFontGetBig5(0xED70) )  
  
  
  
// ■ PRG ■■■■  
  
cMsg := KY_CAPSLK + "■■■■■■"
```

## 5. USRFONT.TTF 預覽功能 (Ctrl+F12)

### 5.1 用途

| 用途       | 說明                                              |
|----------|-------------------------------------------------|
| 驗證造字結果   | 造字後立即在 STK 內看新 glyph 是否正確顯示                     |
| 查 PUA 對映 | 看每個 Unicode codepoint 在 USRFONT.TTF 內的 glyph 樣貌 |
| Debug    | 檢查 FD=4 自我成長後 UMAP 的對映狀態                        |
| 客戶教育     | 展示系統能 render 哪些字                                |

### 5.2 操作 (按下 Ctrl+F12 後)

| 鍵           | 功能                                         |
|-------------|--------------------------------------------|
| Ctrl+F12    | 叫出字型檢視選單                                   |
| 1           | ASCII / ASCFONT Chart (0-255 含控制字元)        |
| 2           | USRFONT.TTF PUA Chart (E000-EE1A, ~3611 字) |
| Esc         | 取消                                         |
| PgDn / PgUp | 上下翻頁 (檢視畫面內)                               |
| Esc         | 離開檢視, 回 Stk32                              |

### 5.3 預覽顯示密度 (USRFONT.TTF 模式)

| 模式                     | 每字佔欄 | 每頁字數 | 總頁數 (3611 字) |
|------------------------|------|------|--------------|
| 密集瀏覽 (純 glyph)         | 2    | 920  | 4            |
| 詳細 (codepoint + glyph) | 8    | 230  | 16           |
| 完整 (含對映 Big5)          | 14   | 115  | 32           |

## 6. FontForge 造字操作指南

### 6.1 取得 FontForge

- 官網：[fontforge.org](http://fontforge.org)（免費開源）
- Windows / Mac / Linux 都有版本

### 6.2 開啟 USRFONT.TTF

| # | 操作                                                |
|---|---------------------------------------------------|
| 1 | FontForge → File → Open → 選 W:\\S002\\USRFONT.TTF |
| 2 | 會顯示完整 glyph table (3611+ 個 PUA glyph)             |
| 3 | 點選欄位上方的 Unicode 範圍輸入框 → 輸入 U+E000 跳到 PUA 區        |

### 6.3 加新 glyph

| # | 操作                                                                       |
|---|--------------------------------------------------------------------------|
| 1 | 找一個空位 codepoint (如 U+EE50 — 看哪個格子是空白方框)                                  |
| 2 | 雙擊該格進入 glyph editor                                                      |
| 3 | 用畫筆工具畫 outline，或複製其他 glyph 來改                                            |
| 4 | 注意 metrics：建議 advance width = 1024 (matches USRFONT.TTF unitsPerEm=1024) |
| 5 | File → Generate Fonts → 選 TrueType → 存回 USRFONT.TTF                      |

### 6.4 注意事項

- 保留 nameID 1/4/6 = "USRFONT-STK"（不要改字型名稱）
- 保留 OS/2 設定不要動 (ulUnicodeRange 等)
- 編輯前先備份 W:\\S002\\USRFONT.TTF
- 完成後在 STK 用 Ctrl+F12 預覽確認

## 7. PUA codepoint 配置慣例

為避免衝突，建議遵循配置慣例：

| 範圍            | 用途                              | 狀態   |
|---------------|---------------------------------|------|
| U+E000 - E03B | 系統小圖示 (esc, f1-f12 等)           | 已用   |
| U+E040 - E1FF | 保留                              | 空閒   |
| U+E200 - E3FF | 建議：客戶資料造字 (罕用字)                 | 空閒可用 |
| U+E400 - EBFF | 保留                              | 空閒   |
| U+EC3F - EDBB | 系統大圖示 (ESC, F1-F12, Ins, Del 等) | 已用   |
| U+EE00 - EE1A | 保留 (USRFONT.TTF 目前涵蓋上限)         | 已部分用 |
| U+EE1B - EFFF | 未來擴充                            | 空閒   |

### 配置原則

1. 系統內建符號 → 用 EC3F-EDBB 區 (高位 PUA)
2. 客戶資料造字 → 用 E200-E3FF 區 (低位 PUA，跟系統符號分開)
3. 同一企業客戶內保持 PUA 配置一致 (每家客戶用獨立子區段)

## 8. 常見問題 Q&A;

| Q                                   | A                                         |
|-------------------------------------|-------------------------------------------|
| 造字後其他客戶也能看到嗎?                       | 看 USRFONT.TTF 部署。若部署到 W:\S002 共享路徑 → 所有客戶 |
| DBF 內存的 Big5 碼會跟 PUA 會codepoint衝突嗎? | DynAllocSlot 自動分配 FA-FE Big5 空位給 PUA      |
| 客戶在 Stk32 內可以自己造字嗎?                 | 目前不行。需要透過外部工具 (FontForge)。未來可考慮在 Stk32    |
| 造的字會被列印出來嗎?                         | 會。GDI 列印用同樣的 USRFONT.TTF 渲染。PDF 預覽列印也同    |
| 如果造字後 Stk32 沒顯示新字怎麼辦?               | (1) 重啟 Stk32 確保載入新 USRFONT.TTF ; (2) Ctrl |
| 同一 PUA codepoint 被兩個客戶造字會衝突。建議怎麼管理? | 建議統一造字管理，或每家客戶獨立 USRFONT.TTF (不           |
| 造的字會自動同步到 USRFONT.UNIAP 嗎?          | 顯示時 DynAllocSlot 自動建立 Big5 Unicode 對      |
| 沒華康字型的機器，造字會正常嗎?                    | 式下會。USRFONT.TTF 自帶向量字，不依賴系統華康。            |

—— 文件結束 ——

有任何問題，先看第 8 章 Q&A;。