# QuickBox GET 內 Ctrl+V 貼上「慢慢貼」效能修復

專案：HBSTK 日期：2026-06-09 檔案：source/GETSYS.PRG

## 一、訴求與現象

在帶有 QuickBox（快速挑選浮窗、會隨輸入做增量過濾）的 GET 欄位（如進銷存的客戶代號、廠商、品名欄）按 Ctrl+V 貼上文字時，字元會一個一個「慢慢」出現，貼一段話要等好幾秒。沒有 QuickBox 的 GET 欄位則貼上飛快。需求：讓 QuickBox 欄位的貼上也能瞬間完成。

## 二、根因分析

貼上功能（PasteHistoryPick／ClipPasteRaw／PasteImageAsText）的注入方式是把整段文字逐位元組 hb_keyPut(Asc()) 塞進鍵盤緩衝區，再交由外層 READ 迴圈逐一消化。而 GETSYS.PRG 在每處理一個字元後，都會執行一次每鍵後掛鉤：

| 關鍵位置 | 內容 | 代價 |
| --- | --- | --- |
| GETSYS.PRG:493 每鍵後掛鉤 | Eval(oGet:cargo[GETINKEYBLOCK], oGet, 1) 即 QuickBox(oGet,1,...)，每個字元觸發一次。 | 與字元數成正比 |
| QuickBox() 本體 | 開啟片語/主檔 DBF、推入滑鼠熱區、SaveScreen、依目前 buffer 掃描建清單、再以 ArrayBrowse 畫出浮窗。 | 單次即昂貴，CJK／造字重繪更慢 |
| 結果 | 貼 N 個字元 = 重建 N 次 QuickBox。 | 視覺上字元一個一個慢慢冒出 |

## 三、踩坑與陷阱

| 陷阱 | 內容 |
| --- | --- |
| 症狀只在特定欄位 | 同樣的 Ctrl+V，無 QuickBox 的 GET 很快、有 QuickBox 的很慢；差別不在貼上本身，而在每鍵後的 QuickBox 刷新掛鉤。 |
| 鍵是「之後」才被消化 | hb_keyPut 只是排進緩衝區，真正逐字處理發生在貼上函式回傳後的外層 READ 迴圈；故無法在貼上函式內「暫時關掉再開」掛鉤，必須在每鍵處理點動手。 |
| 不能誤傷手動打字與半形/全形 | 修法須保證：人手打字（緩衝為空）行為完全不變；BIG5 雙位元組（lead/trail 128-255）中途不可半字刷新；滑鼠移動（負碼）不可誤抑制刷新。 |

## 四、解決方法

在 GetApplyKey 每鍵後刷新 QuickBox 之前，先以 NextKey() 查看鍵盤緩衝區是否還有待處理的「字元鍵或 ENTER」。若還有（代表正在批次貼上或按鍵連發）就略過本次刷新，只在批次最後一鍵（緩衝排空）刷新一次。

| 項目 | 改善前 | 改善後 |
| --- | --- | --- |
| QuickBox 刷新次數 | 每字元 1 次（N 次） | 整批僅最後一鍵 1 次 |
| 守衛條件 | 無 | NextKey() 非（32..255 或 K_ENTER）才刷新 |
| 手動打字 | 每鍵刷新 | 緩衝為空→每鍵刷新（不變） |
| 影響範圍 | — | 一處修改，三條貼上路徑＋連發全受惠 |

效果：貼上瞬間完成，且貼完後 QuickBox 以完整內容正確過濾一次；手動打字即時過濾不受影響。

## 五、使用技術與用法

| 技術 | 用法 |
| --- | --- |
| NextKey()<br>預看緩衝 | NextKey() 取下一個待處理鍵碼但不移除；以此判斷是否仍在批次注入中，達成「時間/批次節流」——只在批次邊界刷新一次。 |
| 字元範圍守衛 | 只在（nNext>=32 .AND. nNext<=255）.OR. nNext==K_ENTER 時抑制刷新，精準對應貼上注入的字元與換行，避開滑鼠（負碼）等事件。 |
| 單點修改全面受惠 | 掛鉤集中在 GETSYS GetApplyKey<br>一處，故所有貼上路徑與按鍵連發自動同享優化。 |
| Python<br>bytes-only 改<br>BIG5 PRG | 讀檔為 bytes、unique 子字串定位 replace、中文註解<br>.encode("big5"）寫入，不 decode 既有內容，避免 UTF-8<br>污染。 |