

PrnAsk 預覽列印快捷鍵 (Alt+P 列印 + Keypad +/- 縮放 + HINT)

日期：2026/06/03 模組：PrnAsk (FiveWin GDI 預覽 / WebView2 預覽)

檔案：PRNASK/source/fw_prnout.prg、PRNASK/source/fw_ime.c (新增)

一、訴求與現象

客戶希望 PrnAsk 的兩種預覽模式 (GDI 視窗、WebView2/pdf.js) 都能用鍵盤快速操作：「Alt+P」直接開列印對話框；鍵盤右側數字鍵盤 (Keypad) 的「+」放大、「-」縮小；並在按鈕上加滑鼠提示 (HINT/Tooltip)。

實際下手後狀況百出：Alt+P

一開始完全無效；改寫法後程式「一執行就閃退」；Keypad +/- 不論怎麼按都「完全沒反應」。看似簡單的快捷鍵，牽出 FiveWin 事件鏈、WebView2 架構、以及最棘手的「中文輸入法 (IME) 攔截鍵盤」三層問題。

二、踩坑與陷阱 (依踩到順序)

#	坑 (症狀)	原因
1	bSysKeyDown 掛在 TWindow 上 → 一執行就閃退 (BASE/1005 Message not found: TWINDOW:_BSYSKEYDOWN)	Alt (WM_SYSKEYDOWN) 要用 bSysKeyDown 攔，但 bSysKeyDown 是 TControl 的 DATA，基類 TWindow 根本沒有這個屬性。DEFINE WINDOW 產生的 oWnd 是 TWindow → 指派即崩潰。
2	WebView2 也閃退 (TWEBVIEW2:_BSYSKEYDOWN)	TWebView2 同樣沒有 bSysKeyDown。而且「WebView2 不是用 FiveWin 寫的」(它是 Chromium+pdf.js 的 HTML/JS)，Harbour 端根本攔不到鍵。
3	WebView2 Alt+P 要先用滑鼠點預覽圖才有效	預覽剛載入時，WebView2 的 child window 沒有取得 Windows 鍵盤焦點，JS 的 keydown 收不到，必須點一下白色預覽區才行。

#	坑（症狀）	原因
4	我加的 oWnd:bKeyDown 把 Keypad +/- 一起害掉	oWnd:bKeyDown 一旦存在就攔下所有 WM_KEYDOWN，原本靠 SetKey(43)/(45) 的 +/- 縮放反而收不到。
5	★核心坑：Keypad +/- 在中文 IME 下被折成 VK 229	裝了中文輸入法時，按 Keypad +/- 送到程式的虛擬鍵碼變成 229 (VK_PROCESSKEY)，代表「IME 正在處理這個鍵」，而且完全不產生 WM_CHAR。所以 bKeyDown 收到的永遠是 229、bKeyChar 連觸發都沒有。
6	ImmAssociateContext 關閉視窗 IME → 無效	對 oWnd:hWnd 解除 IME 關聯後，Keypad 仍是 229。因為按鍵時焦點在子控件（頁碼輸入框、工具列），它們有各自的 IME context，沒被關到。
7	ImmGetVirtualKey 反查原始鍵 → 也回 229	想用 ImmGetVirtualKey 把 229 還原成原始 VK，結果這顆輸入法在 bKeyDown 當下回傳的還是 229，拿不到被折疊前的真實鍵。
8	FiveWin 的 DLLFUNCTION 不能直接用	DLLFUNCTION 是 vrd.prg 內私有的 #command（展開成 DllLoad/DllCall），不在共用 .ch；直接寫會 Syntax error。dll.ch 也沒有 DLL_STDCALL 常數。
9	C 函式 link 失敗：symbol 底線不匹配	C 端寫 HB_FUNC(FW_DISABLEIME) 產生 symbol _HB_FUN_FW_DISABLEIME（有底線）；但 Harbour 端呼叫 FwDisableIME 正規化成 FWDISABLEIME（無底線），link 找不到 _HB_FUN_FWDISABLEIME。
10	bash heredoc 把反斜線吃掉 → hbp 路徑爆掉	在 bash heredoc 內用 Python 寫 Windows 路徑「..\source\fw_ime.c」，反斜線被吞成跳脫字元，「\f」變成 0x0C（form feed），路徑整個壞掉。改用正斜線解決。

#	坑（症狀）	原因
11	C 函式名沒展開（HB_FUNC 被當普通函式）	手動編譯診斷時 hbapi.h 沒 include 到，HB_FUNC 巨集沒展開，obj 裡就沒有 _HB_FUN_ 符號 → 提醒：HB_FUNC 必須在 hbapi.h 之後。

三、解決方法

★ Keypad +/- 的最終正解 —— 讀硬體層鍵盤狀態繞過 IME：

既然 IME 把鍵折成 229、又不給 WM_CHAR、ImmGetVirtualKey 也問不出原始鍵，那就不問 IME。改用 Win32 的 GetAsyncKeyState(VK) 直接讀「硬體層」的按鍵狀態（IME 攔不住硬體層）。當 bKeyDown 收到 229 時，立刻檢查 VK_ADD(107) 與 VK_SUBTRACT(109) 哪一顆物理鍵正被按住，藉此還原使用者按的是 + 還是 -。

項目	做法
GDI：Alt+P 列印	掛在 oBar:bSysKeyDown（buttonbar 是 TControl，有 bSysKeyDown），收到 VK_P(80) 就開列印對話框。
GDI：Keypad +/- 縮放	oWnd:bKeyDown 收到 229 時用 FW_KEYDOWN(107/109)（包 GetAsyncKeyState）判斷物理鍵，呼叫 FwGdiPrvZoom(±25) 縮放（幅度對齊工具列按鈕）。
GDI：ESC / 滑鼠	ESC 關窗用 bKeyDown(27)；右鍵與滾輪縮放維持原樣。
WebView2：Alt+P	純靠 JS keydown 的 e.altKey 判斷（它是 Chromium，不經 Windows IME 229 問題）。
WebView2：自動 focus	oWnd:bInit 時 SetFocus(GetWindow(oWnd:hWnd, GW_CHILD=5))，預覽一出現就把鍵盤焦點帶進 webview，免得要先點一下。
WebView2：Keypad +/-	JS keydown 用 e.code===NumpadAdd / NumpadSubtract 呼叫 zIn()/zOut()。
HINT 提示	GDI 按鈕用 TOOLTIP「放大(Keypad +)」「縮小(Keypad -)」；WebView2 按鈕用 HTML title 屬性（列印(Alt+P)、拉近(I / Keypad +)、拉遠(O / Keypad -)）。

四、使用技術與用法

技術	用法/重點
FiveWin 事件鏈	WM_KEYDOWN→KeyDown→bKeyDown；WM_CHAR→KeyChar→bKeyChar；WM_SYSKEYDOWN(Alt)→SysKeyDown→bSysKeyDown。bSysKeyDown 只在 TControl，bKeyChar/bKeyDown 在 TWindow。
獨立 C 模組	新增 PRNASK/source/fw_ime.c，於 prnask.hbp 以「../source/fw_ime.c」（正斜線）加入。內含 FW_DISABLEIME / FW_KEYDOWN 兩個 HB_FUNC。
動態載入 DLL	C 內用 LoadLibraryA("imm32.dll") + GetProcAddress 動態呼叫，免去 link imm32.lib。GetAsyncKeyState 屬 user32，直接呼叫。
HB_FUNC 命名	Harbour 呼叫端與 C 端 HB_FUNC 名稱「底線要一致」。本案統一為 FW_KEYDOWN / FW_DISABLEIME。
關鍵 VK 碼	VK_ESCAPE=27、VK_P=80、VK_PROCESSKEY=229、VK_ADD(Key pad+)=107、VK_SUBTRACT(Key pad-)=109、GW_CHILD=5。
診斷手法	在 codeblock 內插 WriteDebugLog("GDIkey="+Str(nKey)+...)，從 Debug_PrnAsk.txt 讀實際鍵碼，逐步逼出 229 與硬體層方案。

五、最終成果

GDI 預覽：Alt+P 列印、ESC 關窗、Keypad +/- 縮放(±25%)、放大/縮小鈕 HINT，全部正常。

WebView2 預覽：Alt+P 列印、Keypad +/- 縮放、預覽自動取得焦點、按鈕 HINT、不再閃退。

最大教訓：當鍵盤快捷鍵在中文 IME 環境「完全沒反應」時，先懷疑 IME 把鍵折成 VK_PROCESSKEY(229)；關

IME、反查原始鍵都可能失敗，最可靠的是用 GetAsyncKeyState 讀硬體層繞過 IME。另外，做鍵盤事件前務必先確認該事件屬性是 TWindow 還是 TControl。