

HBSTK 系統修改紀錄

CLIPPER → Harbour GTWVT Console Mode 轉換作業

修改日期：2026-04-11

本次修改摘要

1	列印攔截改檔	18 個 PRG 檔，SET PRINTFILE → OpenSpool/CloseSpool
2	特殊字元常數化	新增 CustomChar.ch，61 個常數，STK.PRG 120 處替換
3	INI 設定系統	新增 stk.ini + ini_loader.prg，48 項設定
4	go32.bat 精簡	移除 35 個 SET 命令，改由 stk.ini 在執行期讀取
5	GETE 全掃描	10 個 PRG 掃描，補齊 6 個缺少的環境變數

編碼規定：所有 PRG 原始碼一律以 BIG5 存檔
終端機：GTWVT BIG5 模式 (HB_CPTERM=BIG5，HB_CODEPAGE=BIG5)
字元輸出：一律使用 Chr(hi)+Chr(lo) BIG5 位元組對，禁止使用 hb_UTF8Chr()

一、列印攔截改檔 (18 個 PRG)

1.1 背景與目的

原 Clipper 程式使用 SET(_SET_PRINTFILE, cFileName) 將列印輸出導向磁碟檔案。
轉換至 Harbour GTWVT 模式後，改為呼叫 compat 層的 OpenSpool / CloseSpool，
由 print_spool.prg 統一管理輸出，並自動以「RPT + 日期 + 時間 + 毫秒 + .PRN」
作為檔名，確保每次列印產生唯一檔案，同時支援 PDF 轉換 (prn_to_pdf.prg)。

1.2 print_spool.prg 關鍵修改

主要新增函式與變數如下：

```

STATIC s_cCurrentPrnFile := "" // 儲存本次列印檔名

FUNCTION GenPrnFileName()
    LOCAL dNow := Date(), cT := StrTran(Time(), ":", "")
    LOCAL nMs := hb_MilliSeconds() % 1000
    RETURN hb_DirBase() + "RPT" + DToS(dNow) + cT + ;
        StrZero(nMs, 3) + ".PRN"

FUNCTION SpoolOpenToFile() // 開啟列印攔截
    s_cCurrentPrnFile := GenPrnFileName()
    SET(_SET_PRINTFILE, s_cCurrentPrnFile)
    // ...

FUNCTION SpoolCloseAndPrint() // 關閉並觸發 PDF 轉換
    IF !Empty(s_cCurrentPrnFile)
        ConvertPrnToPdf(s_cCurrentPrnFile) // prn_to_pdf.prg
    ENDIF

```

1.3 各檔案修改對照表

以下 18 個 PRG 檔案均已備份為 .bak_fileout，並進行以下模式替換：

#	檔案名稱	主要變更
1	P123.PRG	SET PRINTFILE → OpenSpool/CloseSpool (銷售單列印)
2	P221.PRG	SET PRINTFILE → OpenSpool/CloseSpool (採購訂單)
3	P241.PRG	SET PRINTFILE → OpenSpool/CloseSpool (採購入庫)
4	P25.PRG	SET PRINTFILE → OpenSpool/CloseSpool (退貨處理)
5	P246.PRG	SET PRINTFILE → OpenSpool/CloseSpool (入庫明細)
6	P27.PRG	SET PRINTFILE → OpenSpool/CloseSpool (調撥單)
7	P36.PRG	SET PRINTFILE → OpenSpool/CloseSpool (庫存盤點)
8	P351.PRG	SET PRINTFILE → OpenSpool/CloseSpool (庫存明細)
9	P355.PRG	SET PRINTFILE → OpenSpool/CloseSpool (庫存統計)
10	P373.PRG	SET PRINTFILE → OpenSpool/CloseSpool (庫存異動)
11	P415.PRG	SET PRINTFILE → OpenSpool/CloseSpool (應收帳款)
12	P424.PRG	SET PRINTFILE → OpenSpool/CloseSpool (應付帳款)

13	P46.PRG	SET PRINTFILE → OpenSpool/CloseSpool (費用明細)
14	P47.PRG	SET PRINTFILE → OpenSpool/CloseSpool (費用統計)
15	REPORT.PRG	SET PRINTFILE → OpenSpool/CloseSpool (報表主程式)
16	PRINT.PRG	SET PRINTFILE → OpenSpool/CloseSpool (列印工具)
17	P14.PRG	SET PRINTFILE → OpenSpool/CloseSpool (銷貨單)
18	P413.PRG	SET PRINTFILE → OpenSpool/CloseSpool (收款單)

1.4 替換模式說明

每個 PRG 的替換遵循以下統一模式：

```
/* 舊寫法 (Clipper) */  
SET(_SET_PRINTFILE, cFileName) // 開啟  
... 列印邏輯 ...  
SET(_SET_PRINTFILE, "") // 關閉  
  
/* 新寫法 (Harbour compat 層) */  
LOCAL cPrintPort := ""  
OpenSpool( cFileName, , @cPrintPort ) // 開啟攔截  
... 列印邏輯 ...  
CloseSpool( cFileName, , cPrintPort ) // 關閉 → 自動存 PRN + 轉 PDF
```

二、特殊字元常數化 (CustomChar.ch)

2.1 問題背景

原 STK.PRГ 使用 hb_UTF8Chr() 產生圓圈數字①-⑩等特殊符號。
在 GTWVT BIG5 模式下，UTF-8 位元組序列被 BIG5 終端機誤解譯，造成「竈i」等亂碼顯示。

根本解法：一律改用 Chr(hi)+Chr(lo) BIG5 位元組對輸出特殊字元，並以 #define 常數封裝，避免原始碼中出現裸露的 Chr() 數字。

2.2 CustomChar.ch 內容 (新增檔案)

路徑：E:\convertFW\HBSTK\source\CustomChar.ch (BIG5 存檔)
共 61 個常數，分為數字圓圈與功能鍵兩大群組。

2.2.1 UNITRANS 自訂圓圈數字 (UC_* 系列)

用於 user_box_type=true 模式 (UNITRANS 自訂字元區 FC41-FC54)：

常數名稱	BIG5 碼	說明
UC_CIR_1 .. UC_CIR_10	FC41 .. FC4A	UNITRANS 自訂 圓圈①-⑩
UC_PAR_1 .. UC_PAR_10	FC4B .. FC54	UNITRANS 自訂 括號(1)-(10)

2.2.2 標準 BIG5 圓圈數字 (DG_* 系列)

用於標準 BIG5 終端機模式 (C6A1-C6B4)：

常數名稱	BIG5 碼	說明
DG_CIR_1 .. DG_CIR_10	C6A1 .. C6AA	標準 BIG5 圓圈①-⑩
DG_PAR_1 .. DG_PAR_10	C6AB .. C6B4	標準 BIG5 括號(1)-(10)

2.2.3 功能鍵字元 (KY_* 系列)

UNITRANS 自訂功能鍵區 (FB78-FBAF)，共 21 個：

常數名稱	BIG5 碼	代表符號
KY_ALT	FBA7	Alt 修飾鍵
KY_CTRL	FBA8	Ctrl 修飾鍵
KY_INS	FBA3	Insert
KY_DEL	FBA2	Delete
KY_ESC	FB7E	Escape
KY_ENTER1	FB7B	Enter (主鍵盤)
KY_ENTER2	FB7C	Enter (數字鍵盤)
KY_HOME	FB79	Home
KY_HOME2	FB7D	Home (替代)
KY_END	FB78	End
KY_PGUP	FBAB	Page Up

KY_PGDN	FBAC	Page Down
KY_TAB	FBA1	Tab
KY_SEARCH	FBA6	查詢
KY_APPLY	FBAA	確認/F10
KY_MERGE	FBAF	合併/F9
KY_CONVERT	FBA9	轉換
KY_AUDIT	FBAE	稽核
KY_TAXWIN	FBAD	稅務視窗
KY_RIGHT	FBA4	向右 (→)
KY_LEFT	FBA5	向左 (←)

2.3 STK.PRG 修改內容

備份：STK.PRG.bak_adigit

變更：第 5 行新增 #include "CustomChar.ch"。

aDigit[] 與 aPriceLbl[] 陣列共替換 120 處 Chr() 為具名常數。

```
#include "CustomChar.ch"

/* aDigit 陣列 — 依 user_box_type 分支 */
IF GETE("user_box_type") == "true"
  aDigit := { ;                // UNITRANS 自訂字元區
    { UC_CIR_1, UC_CIR_2, ..., UC_CIR_10 }, ; // 圓圈①-⑩
    { UC_PAR_1, UC_PAR_2, ..., UC_PAR_10 }, ; // 括號(1)-(10)
    { DG_CIR_1, DG_CIR_2, ..., DG_CIR_10 }, ; // 標準①-⑩
    { DG_CIR_1, DG_CIR_2, ..., DG_CIR_10 } ; // 標準①-⑩ (備用)
  }
  aPriceLbl := { UC_PAR_1, UC_PAR_2, ..., UC_PAR_10 }
ELSE
  aDigit := { ;                // 標準 BIG5
    { DG_CIR_1, ..., DG_CIR_10 }, ; // 圓圈①-⑩
    { DG_PAR_1, ..., DG_PAR_10 }, ; // 括號(1)-(10)
    ...
  }
ENDIF
```

三、INI 設定系統 (stk.ini + ini_loader.prg)

3.1 設計目的

原系統透過 go32.bat 的 SET 命令注入 35 個環境變數，
導致應用程式環境設定散落於批次檔中，不易維護。

新做法：以 stk.ini 集中管理所有設定，
程式啟動時呼叫 LoadStkIni() 讀取並以 hb_SetEnv() 注入環境，
所有既有的 GETE() 呼叫無需修改即可繼續運作。

3.2 stk.ini 結構 (48 項設定)

路徑：E:\convertFW\HBSTK\build\stk.ini

```
; --- 應用程式控制 ---
TestInkey=
AWI_Home=C:\900
RunCmd=RUN()
CMDHead=START
NET_PRINTER=2
SysIdle=true
SysSet=/OO64 /OU+ /CX4096,0 /OP+ /CE4096,0 /HM+ /EK- /HX+ /HV+
ShowRunCmd=
KeyPosEnable=

; --- 畫面顯示 ---
user_box_type=true
box_type=3
awi_logo=false
background=y
etcolor=2
bgcolor=w
StkCursor=1
C_TITLE=STK

; --- 商業邏輯 ---
BarCodeByPartsId=false,2
ChkAccMonth=3/6
quickrec=30
PriceFirst=nPrice2
RealNoFrom=1
AppendDay=60
CalcStock=
CmpylD=
PlacelD=
now_phs=2
serial=2
status_yn=y

; --- 系統識別 ---
KM=awi
```

```
KMSN=9999-9999-99-99
StartUp=C:\900
clip_date=yyyy.mm.dd

; --- 路徑設定 (支援 %VAR% 替換) ---
sys_drv=E:
sys_path=\ConvertFW\900
exe_drv=E:
exe_path=\ConvertFW\900
loc_sys=%sys_drv%%sys_path%
loc_exe=%exe_drv%%exe_path%
loc_main=%loc_exe%\dbf
loc_dbf=%loc_exe%\dbf
loc_img=%loc_sys%\img
loc_phs=%loc_sys%\phs
loc_rpt=%loc_sys%\rpt
loc_txt=%loc_sys%\txt
loc_frm=%loc_sys%\frm
loc_lbl=%loc_sys%\lbl
loc_tmp=%loc_sys%\tmp
loc_mem=%loc_sys%\mem
loc_set=%loc_sys%\set
```

3.3 ini_loader.prg 核心邏輯

路徑：E:\convertFW\HBSTK\compat\ini_loader.prg (新增檔案)

```

FUNCTION LoadStkIni( cIniFile )
    LOCAL cContent, aLines, cLine
    LOCAL cKey, cVal, nEq
    LOCAL hVars := { => }    // 已載入的變數 Hash
    LOCAL nCount := 0

    IF cIniFile == NIL
        cIniFile := hb_DirBase() + "stk.ini"
    ENDIF

    IF ! File( cIniFile )
        RETURN .F.
    ENDIF

    cContent := hb_MemoRead( cIniFile )
    aLines := hb_ATokens( StrTran(cContent,Chr(13),""), Chr(10) )

    FOR EACH cLine IN aLines
        cLine := AllTrim( cLine )
        IF Empty(cLine) .OR. Left(cLine,1)=";" .OR. ; // 略過空行/註解
            Left(cLine,1)="#" .OR. Left(cLine,1)="["
            LOOP
        ENDIF
        nEq := At( "=", cLine )
        IF nEq == 0 ; LOOP ; ENDIF
        cKey := AllTrim( Left(cLine, nEq-1) )
        cVal := SubStr( cLine, nEq+1 )
        cVal := IniExpandVars( cVal, hVars ) // %VAR% 展開
        hVars[ Upper(cKey) ] := cVal
        hb_SetEnv( cKey, cVal )           // 注入環境
        nCount++
    NEXT
    RETURN .T.

STATIC FUNCTION IniExpandVars( cVal, hVars )
    LOCAL nStart, nEnd, cVarName, cVarVal, nSafety := 0
    DO WHILE nSafety < 20
        nStart := At( "%", cVal )
        IF nStart == 0 ; EXIT ; ENDIF
        nEnd := At( "%", SubStr(cVal, nStart+1) ) // Harbour AT() 僅 2 參數
        IF nEnd == 0 ; EXIT ; ENDIF
        nEnd := nStart + nEnd           // 換算絕對位置
        cVarName := SubStr( cVal, nStart+1, nEnd-nStart-1 )
        cVarVal := hb_HGetDef( hVars, Upper(cVarName), GetEnv(cVarName) )
        cVal := Left(cVal,nStart-1) + cVarVal + SubStr(cVal,nEnd+1)
        nSafety++
    ENDDO
    RETURN cVal

```

重要：Harbour 的 AT() 函式僅接受 2 個參數 (Clipper 支援 3 個) 。
因此以 SubStr() 截斷後再加回偏移量的方式找出關閉 % 的位置。

3.4 main.prg 修改

備份：main.prg.bak_ini

在 InitPrintSpool() 與 STK() 呼叫前插入 LoadStkIni()：

```
// LoadStkIni: 在任何 GETE() 被呼叫之前先讀取 stk.ini
LoadStkIni()
// InitPrintSpool: 載入已儲存的印表機選項
InitPrintSpool()
// 啟動主程式
STK()
```

3.5 stk32.hbp 修改

在 ..\compat\print_spool.prg 之前新增：

```
..\compat\ini_loader.prg
```

四、go32.bat 精簡

4.1 修改說明

備份：go32.bat.bak_ini

移除全部 35 個應用程式層級的 SET 命令，

保留僅與編譯器相關的必要環境設定。

4.2 保留的編譯器環境變數

```
SET DRIVER=E:
set BCC_PATH=%DRIVER%\BCC7732
set HB_PATH=%DRIVER%\Harbour32
set PATH=%BCC_PATH%\bin;%HB_PATH%\bin\win\bcc;%path%
set HB_CPTERM=BIG5
set HB_CODEPAGE=BIG5
set HB_USER_CFLAGS=-I%BCC_PATH%\INCLUDE\windows\crtl ...
set HB_USER_LDFLAGS=-L%BCC_PATH%\LIB;%BCC_PATH%\LIB\psdk
set HB_USER_RESFLAGS=-I%BCC_PATH%\include\windows\sdk -v
```

4.3 移除的應用程式變數 (已移入 stk.ini)

以下 35 個 SET 命令已全部移除，改由執行期 LoadStkIni() 注入：

AWI_Home	IndexPack	RunCmd	SerialBeginNo	Alt5RecCount
Alt6RecCount	CMDHead	Key_F12	NET_PRINTER	SysIdle
SysSet	ShowRunCmd	KeyPosEnable	user_box_type	box_type
awi_logo	background	b_pattern	etcolor	bgcolor
StkCursor	C_TITLE	BarCodeByPartsId	BodySearchByBarCode	ChkAccMonth
quickrec	PriceFirst	RealNoFrom	OpenDbfWait	StkQtyForm
AppendDay	CalcStock	CmpyID	PlaceID	KM

4.4 執行流程說明

建置期：go32.bat 設定編譯器路徑，執行 hbm2 stk32.hbp 完成編譯。

執行期：Stk.exe 啟動時呼叫 LoadStkIni() 讀取 stk.ini，

以 hb_SetEnv() 注入應用程式環境，所有 GETE() 繼續正常運作。

五、GETE() 全域掃描與 stk.ini 補齊

5.1 掃描方法

以 Grep 工具搜尋所有 PRG 檔案中 GETE() 與 GetEnv() 的呼叫，
對照 stk.ini 已有的 KEY 清單，找出缺少的設定項目並補齊。

5.2 掃描結果 (10 個 PRG 檔案)

PRG 檔案	用到的環境變數
STK.PRG	user_box_type, box_type, StkCursor, awi_logo, C_TITLE, etcolor, bgcolor, background, b_pattern, Alt5R
SYSTEM.PRG	SysIdle, KeyPosEnable, ShowRunCmd, SysSet
INITIAL.PRG	AWI_Home, KM, KMSN, StartUp, clip_date
SYSSETUP.PRG	loc_sys, loc_exe, loc_main, loc_dbf, loc_img, loc_phs, loc_rpt, loc_txt, loc_frm, loc_lbl, loc_tmp, loc_mem
APPSETUP.PRG	CmpylID, PlacelD, now_phs, serial, status_yn
REPORT.PRG	loc_rpt
PRINT.PRG	NET_PRINTER, loc_tmp
TRADE.PRG	PriceFirst, BarCodeByPartsId, BodySearchByBarCode, CalcStock
DBF.PRG	OpenDbfWait, loc_dbf
BUYSELL.PRG	quickrec, AppendDay, ChkAccMonth, RealNoFrom, StkQtyForm

5.3 新增至 stk.ini 的項目

以下 6 個變數原本未在 stk.ini 中，掃描後補齊：

變數名稱	預設值	說明
IndexPack		索引重建旗標 (空 = 不重建)
SerialBeginNo		序號起始值
b_pattern		背景圖案 (空 = 純色)
Alt5RecCount	10	Alt+5 最大筆數
Alt6RecCount	10	Alt+6 最大筆數
loc_mem	%loc_sys%\mem	記憶體暫存路徑

六、備份清單

所有修改前的原始檔案均已備份，確認無誤後可統一刪除備份。

#	備份檔案路徑	說明
1	source\STK.PRG.bak_adigit	aDigit/aPriceLbl 修改前
2	source\main.prg.bak_ini	LoadStkIni() 插入前
3	build\go32.bat.bak_ini	移除 SET 命令前
4	source\P123.PRG.bak_fileout	列印攔截修改前
5	source\P221.PRG.bak_fileout	列印攔截修改前
6	source\P241.PRG.bak_fileout	列印攔截修改前
7	source\P25.PRG.bak_fileout	列印攔截修改前
8	source\P246.PRG.bak_fileout	列印攔截修改前
9	source\P27.PRG.bak_fileout	列印攔截修改前
10	source\P36.PRG.bak_fileout	列印攔截修改前
11	source\P351.PRG.bak_fileout	列印攔截修改前
12	source\P355.PRG.bak_fileout	列印攔截修改前
13	source\P373.PRG.bak_fileout	列印攔截修改前
14	source\P415.PRG.bak_fileout	列印攔截修改前
15	source\P424.PRG.bak_fileout	列印攔截修改前
16	source\P46.PRG.bak_fileout	列印攔截修改前
17	source\P47.PRG.bak_fileout	列印攔截修改前
18	source\REPORT.PRG.bak_fileout	列印攔截修改前
19	source\PRINT.PRG.bak_fileout	列印攔截修改前
20	source\P14.PRG.bak_fileout	列印攔截修改前
21	source\P413.PRG.bak_fileout	列印攔截修改前

新增的檔案（無備份需求）

檔案路徑	說明
source\CustomChar.ch	61 個特殊字元常數定義
compat\ini_loader.prg	LoadStkIni() + IniExpandVars()
build\stk.ini	48 項應用程式設定
compat\print_spool.prg	列印攔截核心（已存在，修改版）
compat\prn_to_pdf.prg	PRN 轉 PDF（已存在）

七、統計摘要

統計項目	數值/說明
修改的 PRG 檔案數	21 個 (含 main.prg + STK.PRG + 18 個列印檔)
新增的原始碼檔案	3 個 (CustomChar.ch、ini_loader.prg、stk.ini)
新增的常數定義	61 個 (40 數字 + 21 功能鍵)
Chr() → 常數 替換	120 處 (STK.PRG 內)
列印攔截替換	18 個 PRG，每檔 2-6 處替換
stk.ini 設定項目	48 項 (原 go32.bat 35 項 + 新增 6 項 + 路徑 7 項)
備份檔案總數	21 個 (.bak_adigit / .bak_ini / .bak_fileout)
GETE() 掃描 PRG 數	10 個
補齊的缺少環境變數	6 個
消除的 hb_UTF8Chr()	全部 (改用 Chr(hi)+Chr(lo) BIG5 位元組對)

重要技術規範 (本次確認)

- BIG5 終端機禁用 hb_UTF8Chr()，改用 Chr(hi)+Chr(lo) 位元組對
- Harbour AT() 僅接受 2 個參數，以 SubStr() 偏移方式處理 3 參數場景
- 所有 PRG 原始碼以 BIG5 編碼存檔 (部分檔案含 big5-hkscs 擴充字元)
- stk.ini 的 %VAR% 展開有 20 次迭代安全上限，防止循環參照
- LoadStkIni() 必須在任何 GETE() 呼叫之前執行
- hbm2 編譯期環境 (BCC_PATH、HB_PATH 等) 仍由 go32.bat 設定

本次修改已全部完成，所有備份保留中，確認後可統一清除。