

UpdSubPrice CPACKING 欄位防呆

修 0115 單機版國內進貨 BASE/1003 崩潰
2026-07-23 HBSTK/source/DOMPUR. PRG

一、訴求與現象

客戶 0115(單機版)國內進貨製單時當掉, 錯誤回報:

```
■■ BASE/1003 Variable does not exist: CPACKING
■■■■: UPDSUBPRICE(4223) → IUPARTSINSUB(3985) → DOMARR(389)
      → P614(59) → PULLDOWNMENU → STK → MAIN
```

二、追查與陷阱

#	觀察	結論
1	UPDSUBPRICE/IUPARTSINSUB 在 HBSTK/source 第一次 grep 找不到定義	函式名非在檔名同名檔;真正定義在 DOMPUR. PRG(IUPartsInSub 3868、UpdSubPrice 4124)
2	trace 行號 4223/3985 一度看似與現行源碼有落差	★誤導:實際客戶版行號 4223 = 現行源碼 4223 完全同一份, 不是舊版
3	cPacking 在 UpdSubPrice 有宣告 LOCAL(4126)且從來源 F5 取值	錯的不是區域變數, 而是「值」的來源
4	錯誤行 4223 是 IF Empty((cDestDbf)->cPacking)	★這是存取目的 DBF 的 CPACKING 欄位(不是 memvar)

真兇:‘(cDestDbf)->cPacking’ 存取的是目的 DBF 的 CPACKING 欄位(包裝)。0115 單機版 DBF 結構較舊,‘DomArrS’ 等子檔沒有 CPACKING 欄位(包裝是後加功能,單機版沒跟到欄位遷移),一存取就 BASE/1003。錯誤行 4223 落在‘DomArrS’ 分支(國內進貨子檔)。

三、解決方法(A 防呆)

UpdSubPrice 依目的 DBF 有 4 個分支都會存取

‘(cDestDbf)->cPacking’:EnqSub / DomArrS / ForPurS / ForArrS。在每個‘IF Empty((cDestDbf)->cPacking)...’ 外層加欄位存在守衛:

```
IF (cDestDbf)->(FieldPos("CPACKING")) > 0
  IF Empty( (cDestDbf)->cPacking )
    REPLACE (cDestDbf)->cPacking WITH cPacking
  ELSE
    REPLACE (cDestDbf)->cPacking WITH AllTrim(...)+","+cPacking
  ENDIF
ENDIF
```

‘(cDestDbf)->(FieldPos("CPACKING"))’ 在該工作區內查欄位位置, 0=不存在。缺欄位就整段跳過(包裝資訊暫不記錄, 不崩潰); 有欄位則行為與原本完全相同。所有舊結構單機/客戶一體免疫。

四、後續(B 補欄位, 選做)

若 0115 要實際使用「包裝」功能, 再走 B: 給該站 ‘DomArrS’ (及 EnqSub/ForPurS/ForArrS) DBF 加 ‘CPACKING’ 欄位, 包裝即正常記錄。A 防呆已先止血, B 可日後按需補。

五、使用技術與心法

技術	用途
(alias)->(FieldPos("欄名"))	在指定工作區內判斷欄位是否存在(>0), 跨站 DBF 結構差異防呆的標準做法
錯誤 trace 逐層定位	由 UPDSUBPRICE→IUPARTSINSUB→DOMARR→P614 鎖定操作(國內進貨)與目的 DBF(DomArrS)
Python bytes-only 改 BIG5	4 個相同區塊以精準 bytes replace 包住, 加 BIG5 中文註解, decode 驗證

心法: ‘BASE/1003 Variable does not exist: XXX’ 出現在 ‘(alias)->XXX’ 形式時, 先分辨是「區域變數未宣告」還是「該工作區 DBF 沒有這個欄位」——後者常見於跨站/單機版 DBF 結構不一致, 根治是 FieldPos 守衛或補欄位。另, Harbour 回報完整函式名、Clipper 只認前 10 碼, 對照 trace 時要留意版本差異(本案兩版行號一致=同一份)。