詢價 InqSub.cdx 索引損毀自癒功能報告

日期:2026-07-17 模組:HBSTK / DBF.PRG(NetUse、EnsureCdx、RebuildCdxSafe)

一、訴求與現象

多個客戶站在同一時段內連續崩潰,錯誤訊息一致:

項目	內容
錯誤代碼	DBFCDX/1012 Corruption detected: W:\S001\DBF\InqSub.cdx
觸發動作	進入「詢價(INQUIRY)」畫面
程式追蹤	DBUSEAREA(0) → NETUSE → INQUIRY → APPSETUP → …
影響範圍	多站同時發生,詢價功能完全無法使用

InqSub.cdx 是 InqSub.dbf 的「生產索引(結構化索引,與 DBF 同檔名)」。當索引檔內容損毀,USE 開檔會在「自動開啟同名索引」的當下直接崩潰。

二、踩坑與陷阱

陷阱	說明
既有自癒沒被觸發	原 EnsureCdx 只處理「檔案不存在(!File)」。損毀時檔案「存在」會通過 File() 檢查,卻在開檔崩,自癒條件根本不成立。
崩在 dbSetIndex 之前	生產索引由 USE 自動開啟,錯誤發生在後續 dbSetIndex 之前,無法只在掛索引處防護。
純 BEGIN SEQUENCE 攔不到	Harbour 純 BEGIN SEQUENCE 只攔 explicit BREAK,攔不到 RDD runtime error;必須用 BEGIN SEQUENCE WITH {|e| BREAK(e)} 才能捕捉 DBFCDX/1012。
AUTOPEN 全域關閉有副作用	曾考慮在 NetUse 內 SET AUTOPEN OFF,但會影響「傳空 aIndex 卻靠生產索引自動開」的呼叫者,故僅在損毀復原路徑局部使用。

陷阱	說明
MessageBox 會等按鍵	MessageBox 無選項時一律 InKey() 等待按鍵,不能當自動消失的「請稍候」提示,改用 SAVE/RESTORE SCREEN 自畫框。
DEFAULT 命令未定義	DEFAULT x TO y 在本專案未定義會 E0030,改用 IF x==NIL。

三、解決方法

採低風險 Design B:不更動全域 AUTOPEN,只在 NetUse 加兩道安全網。

步驟	作法
攔截 USE 損毀	USE 外層包 BEGIN SEQUENCE WITH {|e| BREAK(e)};攔到=生產索引損毀 → 暫時 SET AUTOPEN OFF 重開(不自動開損毀索引即可開 DBF)。
依現有機制重建	損毀索引交由下方 dbSetIndex 攔截 → RebuildCdxSafe 在跨站鎖內 FErase + EnsureCdx,依 SET\SubCdx 定義的 tag/鍵重建乾淨索引。
掛索引同步防護	dbSetIndex 亦各自以 WITH-block 包覆,失敗=該 CDX 損毀 → dbClearIndex + RebuildCdxSafe + 再掛,涵蓋「非生產索引」損毀。

立即救援(舊版 exe 亦可,不必等部署):該站全部關閉 Stk32 → 刪除 W:\Sxxx\DBF\InqSub.cdx →
重新啟動。舊版對「缺檔」已能自癒,會自動重建。刪 cdx 是安全的(純索引,靠 dbf 資料 + SubCdx 定義重建,不動資料)。

四、使用技術與用法

技術	用途
BEGIN SEQUENCE WITH bBlock	設定期間錯誤處理區塊,{|e| BREAK(e)} 將 runtime error 導向 RECOVER,捕捉 RDD 損毀例外。
Set(_SET_AUTOPEN)	局部關閉生產索引自動開啟以便重開損毀表;用完立即還原,不影響其他呼叫者。需 #include Set.ch。

技術	用途
FErase / File	刪除損毀的 .CDX;File() 判斷索引是否存在以決定是否重建。
EnsureCdx(既有)	讀 SET\SubCdx 的 CTAGNAME/CTAGKEY/CCONDITION,以 XINDEX 重建 CDX 各 tag。
INDEXEXT	SYSTEM.CH 定義為 ".CDX";aIndex 路徑無副檔名由 RDD 自動補。

五、重建體驗:提示框 + 防多站競爭 + 重顯

索引重建當下再補三個細節,集中於新函式 RebuildCdxSafe(cIdxPath, lCorrupt):

細節	作法
顯示提示	SAVE SCREEN 後自畫置中框「索引重建中, 請稍候…」(用系統既有 cMsgBoxColor / aFrame 配色)。
防多站同時重建	跨站原子鎖:FCreate 暫存檔 + FRename 到 .RIX(目標已存在會失敗=別台在建)。等待時若鎖消失= 別台已建好→自己跳過;逾時約 20 秒(0.3×67)視為殘鎖強制接手。暫存名以 NetName() 站台專屬避免撞名。
重顯	重建結束 RESTORE SCREEN,把重建前的畫面/提示原樣還原。

六、★真正根因:自癒重建的 tag 順序錯亂

索引 tag 本應 DBFNAME1, 2, … X 依序,卻常變成 DBFNAMEX, 1, 2 …。因全系統用 dbSetOrder(1/2) 靠「順序號」選索引(23 處,無一用 tag 名稱),順序一亂 → dbSetOrder(1) 抓錯索引 → 錯誤 seek / 寫錯位置 → 資料異常,甚至演變成 1012 損毀。這極可能才是索引反覆損毀的上游根因。

查證	結果
DBFCDX 的 ORDER 號依據	自寫測試證實=tag「建立順序」,關閉重開穩定(非 字母重排)。
SubCdx.DBF 實體順序	本身就亂:ActSub 存成 ActSub2→ActSub1;ActMain 存成 1,3,2,4,5。

查證	結果
EnsureCdx 走法	dbGoTop + dbSkip 實體序線性掃描(autopen 雖開 SUBCDX.CDX 但 IndexOrd()=0 走自然序),照亂序建立 → ORDER 1=ActSub2(錯)。
IndexDbf 走法	dbSeek + 索引 UPPER(CCDXNAME+CTAGNAME) → ActSub1, ActSub2(正確)。差別=帶不帶索引走訪。
修法	EnsureCdx 收集後加 ASort,依 Upper(tag 名)由小到大排序,重現索引鍵順序,與 IndexDbf 一致;不改用 dbSeek 以免依賴同時可能損毀的 SubCdx.cdx。
實測	ActSub[2,1]→[1,2];ActMain[1,3,2,4,5]→[1,2,3,4,5]。

七、修改與驗證

項目	內容
修改檔案	E:\convertFW\HBSTK\source\DBF.PRG(NetUse、新增 RebuildHint / RebuildCdxSafe)
備份	DBF.PRG.bak_cdxheal_20260717_1520、bak_cdxhint_20260717_1600、bak_tagsort_20260717_1945
編碼	BIG5,中文註解以 Python bytes-only 寫入
建置	hbmk2 stk32.hbp -comp=bcc → BUILD OK,無 DBF.PRG 編譯警告
冒煙測試	Stk32.exe 正常啟動並存活
未實地驗證	損毀路徑需站上有損毀 cdx + 詢價資料環境;本機 W: 不可達無法完整跑該流程。缺檔與損毀共用同一 RebuildCdxSafe,建議測試站刪一個 cdx 觀察提示框與重建。
待部署	新版 exe 需經更新管線派送至各站;派送前各站可用「立即救援」先恢復。