Probleme beim Einsatz von CurrentDb vermeiden
Versionen: Access 2013, 2010, 2007, 2003 und 2002/XP

Mit der Funktion "CurrentDb()" initialisieren Sie regelmäßig Objektvariablen, um damit bequem auf Objekte, Methoden oder Daten der aktuellen Datenbank zugreifen zu können. Das kann manchmal allerdings zu scheinbar unerklärbaren Laufzeitfehlern und stundenlanger Fehlersuche führen. Unser Tipp zeigt, wie Sie diese Probleme in Zukunft vermeiden.

Beginnen wir mit einem Beispiel. In vielen VBA-Routinen setzen Sie die Funktion "CurrentDb()" ein, um schnell ein Recordset zu initialisieren:

Set rs = \_
  CurrentDB.OpenRecrodset("Select \* from Kunden", \_
                          dbOpenDynaset)

Darüber hinaus gibt es noch verschiedene andere Aufgaben, die sich mit "CurrentDb()" vermeintlich schnell und einfach erledigen lassen. Das Verfahren birgt jedoch eine Gefahr: Bei jedem ersten Zugriff auf eine Tabelle oder eine Abfrage belegt Access einen Handle für die einfachere Verwaltung weiterer Zugriffe. Dieser Handle wird normalerweise bei einem "Recordset.Close" mit anschließendem "Set rs = Nothing" wieder freigegeben. Erfolgt der Zugriff nun über "CurrentDb()", wird kein "Close" ausgeführt und auch keine Dereferenzierung mit "= Nothing" durchgeführt, so dass das Handle nicht wieder freigegeben wird. Wenn Sie also in einer komplexen Datenbank ausgiebig mit "CurrentDb()" arbeiten, wird früher oder später ein Laufzeitfehler 3048, "Es können keine weiteren Datenbanken geöffnet werden!" angezeigt.

Reproduzieren lässt sich das Problem mit einer Testprozedur, die eine größere Anzahl von Recordsets per "CurrentDb()" initialisiert. Kopieren Sie die folgende Prozedur in ein Modul Ihrer Datenbank und starten Sie sie per **F5** oder Klick auf das *Wiedergabe*-Symbol:

Sub CurrentDB\_StressTest()
  Dim I As Long, Cnt As Long
  Dim arrRS(10000) As DAO.Recordset

  For I = 1 To 10000
    Set arrRS(I) = \_
      CurrentDb.OpenRecordset("Select \* from Kunden", \_
                              dbOpenDynaset)
    arrRS(I).MoveLast
    Cnt = arrRS(I).RecordCount
  Next I

End Sub

Je nach Access-Version und Komplexität der Datenbank, in der die Prozedur läuft, wird der Laufzeitfehler 3048 nach ca. 150 bis 250 Aufrufen von "CurrentDb()" angezeigt.

Die Lösung für dieses Problem ist relativ einfach:

* Deklarieren Sie in einem Modul eine globale Variable mit Namen "gDB" oder ähnlich vom Datentyp "DAO.Database":
* ublic gDB As DAO.Database
* Legen Sie eine Funktion an, die diese Variable initialisiert:

Function Init\_gDB()

  Set gDB = CurrentDb()

End Function

* Erstellen Sie ein Makro "AutoExec" mit einer Aktion "AusführenCode" und dem Aufruf dieser Funktion oder fügen Sie eine entsprechend Aktion zu einem vorhandenen "AutoExec"-Makro hinzu.
* Nutzen Sie die Suchen-/Ersetzen-Funktion der VBA-Entwicklungsumgebung, um in Ihren Routinen die Zeichenfolge "CurrentDb." gegen die Zeichenfolge "gDB." auszutauschen.

Durch dieses Änderung besteht nun keine Gefahr mehr, dass ein Laufzeitfehler 3048 angezeigt wird, denn es gibt nur eine einzige Referenz, die Sie jeweils durch Verwendung der globalen Variablen "gDB" nutzen.