

Re: Einzelschritt vs. Komplettdurchlauf

Source: <http://www.tech-archive.net/Archive/German/VB/microsoft.public.de.vb/2007-11/msg00373.html>

- *From:* "Peter Götz" <gssg_nospam@xxxxxxxxxxx>
 - *Date:* Tue, 20 Nov 2007 11:53:07 +0100
-

Hallo Oliver,

ich hab ein Phänomen, dem ich nicht auf die Schliche komme: Ich prüfe in einer Prozedur, ob eine bestimmte Datei geöffnet ist oder nicht. Wenn die Datei offen ist, wird eine Schleife durchlaufen, bis die Datei geschlossen wurde. Dann geht es weiter.

Wenn ich die Prozedur zzum Testen schrittweise mit F8 durchlaufe, klappt alles wie es soll.

Weil Windows dann genügend Zeit zwischen den einzelnen Schritten genügend Zeit hat, alle anstehenden Anforderungen/Nachrichten abzuarbeiten.

Wird die Prozedur in einem Rutsch abgearbeitet,

.... dann bekommt Windows die o.g. Zeit nicht um anstehende Anforderungen/Nachrichten abzuarbeiten.

tut sie so, wie wenn die Datei nicht offen

wäre!?!?!?

Ich habe im Folgenden Deinen Code in eine halbwegs lesbare Form gebracht.

Re: Einzelschritt vs. Komplettdurchlauf

Hier der Code:

```
*****
```

```
Sub Ist_Datei_offen(var_Dateiname)
```

```
Dim var_File As Integer
```

```
On Error Resume Next
```

```
Do
```

```
Err.Clear
```

```
var_File = FreeFile
```

```
Open var_Dateiname _
```

```
For Binary Access Read Lock Read As #var_File
```

Das nachfolgende Close #var_File macht nur dann Sinn, wenn die Datei mit dem soeben ausgeführten Open wirklich geöffnet werden konnte.

War die Datei bereits anderweitig geöffnet, kann sie hier nicht nochmal geöffnet werden und Dein nachfolgendes Close ist sinnlos.

```
Close #var_File
```

```
Select Case Err.Number
```

```
Case 0
```

```
Case 70
```

```
MsgBox var_Dateiname + _
```

```
"Datei ist bereits offen! " & _
```

"Bitte schließen und auf OK klicken."

Strings verkettet man mit dem "&"-Operator. Es funktioniert zwar auch mit dem "+"-Operator aber dabei kann es zu unliebsamen u. unerwarteten Ergebnissen kommen.

```
Case Else: MsgBox "Unbekannter Fehler"
```

```
End Select
```

```
MsgBox Err.Number 'nur zum Test --> später raus damit
```

Re: Einzelschritt vs. Komplettdurchlauf

Loop Until Err.Number = 0
'in der Schleife bleiben, bis die Datei geschlossen ist

End Sub

Wäre toll, wenn mir jemand von euch sagen könnte,
wo der Haken sich
versteckt.

Soweit ich aus Deiner etwas nebulösen Beschreibung
herauslesen konnte, möchtest Du irgendwelchen Code
starten, nachdem eine bestimmte Datei geöffnet und
anschliessend wieder geschlossen worden ist.

Die gesamte Ablauflogik Deines Codes ist nicht
unbedingt das, was zur Lösung einer solchen
Aufgabenstellung notwendig wäre.

Übrigens läuft das alles unter Office 2000 –
falls es ne Rolle spielen sollte.

Nein, das spielt keine Rolle

Schau Dir mal das nachfolgende Beispiel an.

```
'/// Code in einer leeren Form1
Option Explicit
Private WithEvents cmdOpen As CommandButton
Private WithEvents cmdClose As CommandButton
Private WithEvents mTimer As Timer

Private mFileNumber As Integer

' ***** Dateiname anpassen *****
Private Const mFileName = "D:\Testdatei.dat"

Private Sub Form_Load()
Me.ScaleMode = vbPixels
Me.Move 100, 100, _
Me.ScaleX(270, vbPixels, vbTwips), _
Me.ScaleY(150, vbPixels, vbTwips)

CreateControls
End Sub

Private Sub Form_QueryUnload _
```

Re: Einzelschritt vs. Komplettdurchlauf

Re: Einzelschritt vs. Komplettdurchlauf

(Cancel As Integer, UnloadMode As Integer)

```
If mFileNumber > -1 Then
Close mFileNumber
End If
mTimer.Enabled = False
End Sub
```

```
Private Sub cmdOpen_Click()
Dim FN As Integer
FN = FreeFile
```

```
On Error Resume Next
Open mFileName For Binary Access Read Lock Read As #FN
```

```
Select Case Err.Number
Case 0
' Datei konnte geöffnet werden
mTimer.Enabled = False
cmdClose.Enabled = True
mFileNumber = FN
Me.Caption = Me.Name
```

```
Case 70
' Zugriff verweigert
MsgBox "Datei" & vbCrLf & _
mFileName & vbCrLf & _
"ist bereits anderweitig geöffnet!" & _
vbCrLf & vbCrLf & _
"Schliessen sie erst die Datei!", _
vbExclamation
```

```
Case Else
MsgBox "Fehler: " & CStr(Err.Number) & _
vbCrLf & _
"Source: " & Err.Source & _
vbCrLf & _
Err.Description, _
vbCritical
End Select
End Sub
```

```
Private Sub cmdClose_Click()
Close #mFileNumber
cmdClose.Enabled = False
With mTimer
.Interval = 1000
.Enabled = True
End With
mFileNumber = -1
ShowTime
```

Re: Einzelschritt vs. Komplettdurchlauf

```
End Sub
```

```
Private Sub mTimer_Timer()  
ShowTime  
End Sub
```

```
Private Sub ShowTime()  
Me.Caption = Format$(Now, "hh:nn:ss")  
End Sub
```

```
Private Sub CreateControls()  
Set cmdOpen = _  
Me.Controls.Add("VB.CommandButton", "cmdOpen")
```

```
With cmdOpen  
.Font.Name = "Arial"  
.Font.Size = 10  
.Move 10, 10, 120, 36  
.Caption = "Datei öffnen"  
.Visible = True  
End With
```

```
Set cmdClose = _  
Me.Controls.Add("VB.CommandButton", "cmdClose")
```

```
With cmdClose  
Set .Font = cmdOpen.Font  
.Move 135, 10, 120, 36  
.Caption = "Datei schliessen"  
.Visible = True  
.Enabled = False  
End With
```

```
Set mTimer = _  
Me.Controls.Add("VB.Timer", "mTimer")  
With mTimer  
.Enabled = False  
End With  
End Sub  
' \\ E N T E
```

Nach dem Programmstart die Datei über den Button "Datei öffnen" öffnen.

Versucht der Benutzer nun die Datei ein weiteres mal zu öffnen, erhält er eine MsgBox mit der Aufforderung die Datei erst zu schliessen.

Wird die Datei geschlossen dann wird damit der nachfolgende Code (Start eines Timers, der die Uhrzeit in der Titelleiste anzeigt) aufgerufen.

Re: Einzelschritt vs. Komplettdurchlauf

Wenn Du eine fortlaufende Überwachung darüber ob Deine Datei geöffnet oder geschlossen ist, dann kannst Du das ohne blockierende Schleife einfach mit einem Timer erreichen.

```
' /// Code in einer leeren Form1
Option Explicit
Private WithEvents cmdOpen As CommandButton
Private WithEvents cmdClose As CommandButton
Private WithEvents mTimer As Timer

Private mFileNumber As Integer

' ***** Dateiname anpassen *****
Private Const mFileName = "D:\Testdatei.dat"

Private Sub Form_Load()
Me.ScaleMode = vbPixels
Me.Move 100, 100, _
Me.ScaleX(270, vbPixels, vbTwips), _
Me.ScaleY(150, vbPixels, vbTwips)

mFileNumber = -1
CreateControls

With mTimer
.Interval = 100
.Enabled = True
End With
End Sub

Private Sub Form_QueryUnload _
(Cancel As Integer, UnloadMode As Integer)

If mFileNumber > -1 Then
Close mFileNumber
End If
mTimer.Enabled = False
End Sub

Private Sub cmdOpen_Click()
Dim FN As Integer
FN = FreeFile

On Error Resume Next
Open mFileName For Binary Access Read Lock Read As #FN

Select Case Err.Number
Case 0
' Datei konnte geöffnet werden
cmdClose.Enabled = True
```

Re: Einzelschritt vs. Komplettdurchlauf

```
mFileNumber = FN
```

```
Case Else  
MsgBox "Fehler: " & CStr(Err.Number) & _  
vbCrLf & _  
"Source: " & Err.Source & _  
vbCrLf & _  
Err.Description, _  
vbCritical  
End Select  
End Sub
```

```
Private Sub cmdClose_Click()  
Close #mFileNumber  
cmdClose.Enabled = False  
mFileNumber = -1  
End Sub
```

```
Private Sub mTimer_Timer()  
If mFileNumber > -1 Then  
Me.Caption = "Datei ist geöffnet"  
Else  
Me.Caption = "Datei ist geschlossen"  
End If  
End Sub
```

```
Private Sub CreateControls()  
Set cmdOpen = _  
Me.Controls.Add("VB.CommandButton", "cmdOpen")
```

```
With cmdOpen  
.Font.Name = "Arial"  
.Font.Size = 10  
.Move 10, 10, 120, 36  
.Caption = "Datei öffnen"  
.Visible = True  
End With
```

```
Set cmdClose = _  
Me.Controls.Add("VB.CommandButton", "cmdClose")
```

```
With cmdClose  
Set .Font = cmdOpen.Font  
.Move 135, 10, 120, 36  
.Caption = "Datei schliessen"  
.Visible = True  
.Enabled = False  
End With
```

```
Set mTimer = _  
Me.Controls.Add("VB.Timer", "mTimer")
```

Re: Einzelschritt vs. Komplettdurchlauf

Re: Einzelschritt vs. Komplettdurchlauf

```
With mTimer  
.Enabled = False  
End With  
End Sub  
'\\ E N T E
```

Gruß aus St.Georgen
Peter Götz
www.gssg.de (mit VB-Tipps u. Beispielprogrammen)