



amvExplorer mit Behandlung der Ereignisse durch Dateiänderungen

André Minhorst - 2024-03-10 - Kommentare (0) - amvExplorer

amvExplorer mit Behandlung der Ereignisse durch Dateiänderungen

Gegebenenfalls möchtest Du nicht nur einfach ein Verzeichnis anzeigen, sondern auch die darin enthaltenen Dateien in irgendeiner Form in der Datenbank erfassen – beispielsweise durch Verknüpfung auf die jeweiligen Pfade, die Du in einer Tabelle speicherst.

amvExplorer kann alle Änderungen am aktuellen Verzeichnis erfassen:

- Hinzufügen von Dateien
- Umbenennen von Dateien
- Verschieben/Löschen von Dateien
- Hinzufügen von Ordnern
- Umbenennen von Ordnern
- Verschieben/Löschen von Ordnern

Wie das gelingt, zeigen wir im letzten Beispiel. Dieses sieht in der Entwurfsansicht wie folgt aus:

Es enthält neben den bisher bekannten Elementen noch zwei Schaltflächen und ein Unterformular:

- Die erste Schaltfläche startet den Watcher, also die Funktion, die Änderungen am aktuell angezeigten Verzeichnis erfasst.
- Die zweite Schaltfläche beendet den Watcher.
- Das Unterformular zeigt die Daten der Tabelle **TBL_AMV_FILE** an.

Wir deklarieren wieder **objExplorer** und eine Variable für den aktuellen Pfad:

```
Private WithEvents objExplorer As CLASS_AMV_EXPLORER
Private sInitializePath As String
```

In **Form_Open** fügen wir eine Anweisung hinzu, die den Timer aktiviert:

```

Private Sub Form_Open(Cancel As Integer)On Error Resume Next
sInitializePath = CurrentProject.Path & "\FOLDER" Set objExplorer =
New CLASS_AMV_EXPLORER With objExplorer .Initialize
Me.ctlExplorer, sInitializePath, Me.CMD01, Me.CMD02, Me.LBPATH End
With Me.TimerInterval = 100 ' objExplorer.WatchStart --> must be
activated via the timer interval so that the web control is readyEnd
Sub

```

Die Timer-Prozedur startet den Watch-Mechanismus:

```

Private Sub Form_Timer()On Error Resume Next Me.TimerInterval = 0
objExplorer.WatchStart ' Must be set to start the triggerEnd Sub

```

Mit der ersten Schaltfläche starten wir den Watcher:

```

Private Sub CMD03_Click() Me.ctlExplorer.SetFocus
objExplorer.WatchStartEnd Sub

```

Mit der zweiten beenden wir diesen:

```

Private Sub CMD04_Click() Me.ctlExplorer.SetFocus
objExplorer.WatchStopEnd Sub

```

objExplorer bietet ein Ereignis, das bei jeder Änderung an einer Datei ausgelöst wird und das wir wie folgt implementiert haben. Hier werden die Änderungen in die Tabelle **TBL_AMV_FILE** geschrieben.

Der Parameter **sFolderPath** liefert den aktuellen Ordner, **sFileName** die geänderte Datei oder den geänderten Ordner und **stcEventAction** die Aktion, die durchgeführt werden soll. Diese werden entsprechend ausgewertet:

```

Private Sub objExplorer_WatchEvent(ByVal sFolderPath As String, ByVal
sFileName As String, _
    stcEventAction As Long)
Dim sArea As String
Dim sPath As String
Dim sFileAction As String
Static sTmpName As String
On Error Resume Next
Select Case stcEventAction
Case FILE_ACTION_MODIFIED
sFileAction = "MODIFIED"
Case FILE_ACTION_ADDED
sFileAction = "ADDED"
Case FILE_ACTION_REMOVED
sFileAction = "REMOVED"

```

```

        Case FILE_ACTION_RENAMED_NEW_NAME
            sFileAction = "RENAME NEW"
        Case FILE_ACTION_RENAMED_OLD_NAME
            sFileAction = "RENAME OLD"
    End Select

    sPath = sFolderPath
    If Right(sPath, 1) <> "\" Then sPath = sPath & "\"
    If Len(sFileName) > 0 Then
        If sFileAction = "REMOVED" Then
            sArea = "..."
        Else
            If IsFolder(sPath & sFileName) = True Then
                sArea = "DIR"
            Else
                sArea = "FILE"
            End If
        End If
        If stcEventAction = FILE_ACTION_RENAMED_OLD_NAME Then
            sTmpName = Replace(sFileName, "'", "'")
        ElseIf stcEventAction = FILE_ACTION_RENAMED_NEW_NAME Then
            sFileName = Replace(sFileName, "'", "'")
            sFileAction = "RENAME OLD"
            CurrentDb.Execute "INSERT INTO TBL_AMV_FILE (FEVENT,
FAREA, FPATH, FDATE TIME) VALUES('" & sFileAction & "', '" & sArea &
"', '" & sTmpName & "', #" & Format(Now(), "dd\mm\yyyy hh:mm:ss") &
"#)"
            sFileAction = "RENAME NEW"
            CurrentDb.Execute "INSERT INTO TBL_AMV_FILE (FEVENT,
FAREA, FPATH, FDATE TIME) VALUES('" & sFileAction & "', '" & sArea &
"', '" & sFileName & "', #" & Format(Now(), "dd\mm\yyyy hh:mm:ss")
& "#)"
            Forms("FRM_AMV_EXPLORER")("LIST01").Form.Requery
        Else
            sFileName = Replace(sFileName, "'", "'")
            CurrentDb.Execute "INSERT INTO TBL_AMV_FILE (FEVENT,
FAREA, FPATH, FDATE TIME) VALUES('" & sFileAction & "', '" & sArea &
"', '" & sFileName & "', #" & Format(Now(), "dd\mm\yyyy hh:mm:ss")
& "#)"
            Forms("FRM_AMV_EXPLORER")("LIST01").Form.Requery
        End If
    End If

```

End If

End Sub

Außerdem gibt es noch zwei Ereignisse, die beim Starten und Beenden des Watchers ausgelöst werden und die wir wie folgt implementieren:

```
Private Sub objExplorer_WatchTriggerStart()  
    Me.ctlExplorer.SetFocus  
    CMD03.Enabled = False  
    CMD04.Enabled = True  
End Sub
```

```
Private Sub objExplorer_WatchTriggerStop()  
    Me.ctlExplorer.SetFocus  
    CMD03.Enabled = True  
    CMD04.Enabled = False  
End Sub
```

Hinweis

Achtung: Die für diese Anleitung nötige Beispieldatenbank erhältst Du nach dem Kauf des **amvExplorers**:

[Zum Shop...](#)

Hast Du den **amvExplorer** bereits gekauft, findest Du diesen nach der Anmeldung im Shop unter **Mein Konto|Sofortdownloads**.