

Das FileSystem-Objekt

Autor & Copyright: Dipl.-Ing. Harald Nahrstedt

Version: 2016 / 2019 / 2021 / 365

Erstellungsdatum: 09.09.2022

Überarbeitung: 22.02.2024

Beschreibung: Das FileSystem-Objekt ermöglicht den Zugriff auf das Dateisystem eines Computers.

Anwendungs-Datei: AW-002_FileSystemObject.docm

1 Syntax

Das *FileSystem*-Objekt ermöglicht den Zugriff auf das Dateisystem eines Computers und ist ein Objekt der *Scripting Runtime* Bibliothek. Damit es bei einer frühen Bindung verwendet werden kann, muss in der Entwicklungsumgebung unter Register EXTRAS / VERWEISE die *Microsoft Scripting Runtime* Bibliothek eingebunden werden.

Die Syntax des Objekts lautet

```
Scripting.FileSystemObject.
```

Im Wesentlichen stellt das *FileSystem*-Objekt drei Objekte zur Verfügung:

- Drive
- Folder
- File.

Diese korrespondieren mit den Objektlisten

- Drives
- Folders
- Files.

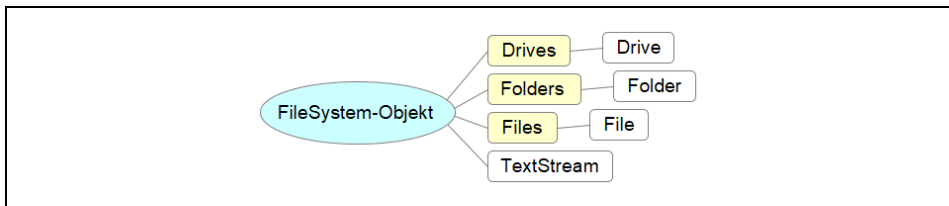


Bild 1. Das *FileSystem*-Objekt und seine Unterobjekte

Die folgende Prozedur liefert Informationen zu allen vorhandenen Laufwerken (Drives).

Codeliste 1. Die Prozedur liefert Informationen zu allen vorhandenen Laufwerken

```
Sub DrivesInfo()
  Dim fsoObj As New FileSystemObject
  Dim drvTmp As Drive
  Dim dvsTmp As Drives

  Set dvsTmp = fsoObj.Drives
  For Each drvTmp In dvsTmp
    'Laufwerksbuchstabe
    With drvTmp
      Debug.Print .DriveLetter,
      If .IsReady Then
        'Laufwerkstyp
        If .DriveType = Remote Then
          Debug.Print .ShareName,
        Else
          Debug.Print .VolumeName,
        End If
      'Speicherplatz
      Debug.Print .FreeSpace & " von " & _
        .TotalSize & " frei"
    End With
  Next
End Sub
```

```

        Debug.Print "Keine CD eingelegt"
    End If
End With
Next
Set dvsTmp = Nothing
Set fsoObj = Nothing
End Sub

```

Die Prozedur prüft alle vorhandenen Laufwerke. Zuerst wird er Laufwerksbuchstabe ausgegeben. Dann wird mit der Eigenschaft `isReady` geprüft, ob das Laufwerk aktiv ist. CD-Laufwerke geben hier ein `False` zurück, wenn keine CD eingelegt ist. Danach wird der Laufwerksname angegeben. Vorher wird jedoch über die Eigenschaften `DriveType` der Laufwerkstyp geprüft, da für Netzwerke der Name mit der Eigenschaft `ShareName` und ansonsten mit `VolumeName` abgefragt werden muss. Zum Schluss folgt die Speicherplatzangabe.

Bei Wechseldatenträgern ist die Eigenschaft `SerialNumber` noch wichtig, mit der eindeutig ein Datenträger aufgerufen werden kann. Das ist hilfreich, wenn die Wechseldatenträger den gleichen Laufwerksbuchstaben besitzen.

Analog zu den Laufwerken zeigt die folgende Prozedur Informationen zu den Verzeichnissen eines Laufwerks oder eines übergeordneten Verzeichnisses an.

Codeliste 2. Die Prozedur liefert Informationen zu allen vorhandenen Verzeichnissen

```

Sub FoldersInfo()
    Dim fsoObj As New FileSystemObject
    Dim fosTmp As Folders
    Dim folTmp As Folder
    Dim sFolder As String

    sFolder = "C:\Temp"
    'Prüfen, ob Verzeichnis existiert
    If fsoObj.FolderExists(sFolder) Then
        Set folTmp = fsoObj.GetFolder(sFolder)
        Set fosTmp = folTmp.SubFolders

        For Each folTmp In fosTmp
            Debug.Print folTmp.DateCreated; folTmp.Attributes; _
                folTmp.Name, folTmp.Path
        Next
        Set folTmp = Nothing
        Set fosTmp = Nothing
    End If
End Sub

```

Die Methode `FolderExists` prüft die Existenz des Verzeichnisses. Anschließend werden die Daten der Unterverzeichnisse ausgegeben. Neben der Eigenschaft `DateCreated` gibt es noch `DateLastAccessed` und `DateModified`. Danach folgen verschiedene Eigenschaften bitweise ausgelesen werden, die nicht nur für Verzeichnisse, sondern auch für Dateien gelten.

Die folgende Prozedur zeigt den Zugriff auf Dateien.

Codeliste 3. Die Prozedur liefert Informationen zu allen vorhandenen Dateien

```
Sub FilesInfo()  
    Dim fsoObj As New FileSystemObject  
    Dim folTmp As Folder  
    Dim flsTmp As Files  
    Dim filTmp As File  
    Dim sStart As String  
  
    sStart = "C:\Temp"  
    If fsoObj.FolderExists(sStart) Then  
        Set folTmp = fsoObj.GetFolder(sStart)  
        Set flsTmp = folTmp.Files  
        For Each filTmp In flsTmp  
            Debug.Print filTmp.DateCreated; filTmp.Attributes, _  
                filTmp.Size, filTmp.Name, filTmp.Type  
        Next  
        Set folTmp = Nothing  
        Set flsTmp = Nothing  
    End If  
End Sub
```

Auch hier wird zuerst die Existenz des Verzeichnisses geprüft.

Im folgenden Beispiel wird das *FileSystem-Objekt* verwendet, um ein *TextStream*-Objekt zu erhalten, aus dem gelesen bzw. in das geschrieben werden kann. Die folgende Prozedur wird in Word im *ThisDocument*-Objekt oder in Excel im *DieseArbeitsmappe*-Objekt erstellt. Hier habe ich Word verwendet.

Codeliste 4. Die Prozedur schreibt einen Text mithilfe eines TextStreams in eine Textdatei

```
Option Explicit  
  
Sub CreateTextStream()  
    Dim fsoObj As FileSystemObject  
    Dim tsmTxt As TextStream  
  
    Set fsoObj = CreateObject("Scripting.FileSystemObject")  
    Set tsmTxt = fsoObj.CreateTextFile("C:\Temp\Test.txt", True)  
    tsmTxt.WriteLine ("Dies ist ein Testtext.")  
    tsmTxt.Close  
    Set tsmTxt = Nothing  
    Set fsoObj = Nothing  
End Sub
```

Die Methode *CreateObject* erstellt und gibt einen Verweis auf ein *ActiveX*-Objekt zurück. Die Methode *CreateTextFile* erstellt ein *TextStream*-Objekt der angegebenen Datei, mit dem aus der Datei gelesen oder in die Datei geschrieben werden kann. Die Methode *WriteLine* schreibt eine bestimmte Zeichenfolge und ein Zeilenumbruchzeichen in die *TextStream*-Datei. Die Methode *Close* leert den *Stream* und schließt ihn.

2 Methoden

Tabelle 1. Die Methoden des *FileSystemObject*-Objekts

| Methode | Beschreibung |
|-----------|--|
| BuildPath | Fügt einen Namen an einen vorhandenen Pfad an. |

| Methode | Beschreibung |
|---------------------|---|
| CopyFile | Kopiert eine oder mehrere Dateien von einem Speicherort an einen anderen. |
| CopyFolder | Kopiert einen oder mehrere Ordner von einem Speicherort an einen anderen. |
| CreateFolder | Erstellt einen neuen Ordner. |
| CreateTextFile | Erstellt eine Textdatei und gibt ein TextStream-Objekt zurück, das zum Lesen aus oder zum Schreiben in die Datei verwendet werden kann. |
| DeleteFile | Löscht eine oder mehrere der angegebenen Dateien. |
| DeleteFolder | Löscht einen oder mehrere der angegebenen Ordner. |
| DriveExists | Überprüft, ob ein angegebenes Laufwerk vorhanden ist. |
| FileExists | Überprüft, ob eine angegebene Datei vorhanden ist. |
| FolderExists | Überprüft, ob ein angegebener Ordner vorhanden ist. |
| GetAbsolutePathName | Gibt den vollständigen Pfad von der Stammebene des Laufwerks für den angegebenen Pfad zurück. |
| GetBaseName | Gibt den Basisnamen einer bestimmten Datei oder eines bestimmten Ordners zurück. |
| GetDrive | Gibt ein Laufwerksobjekt zurück, das dem Laufwerk an einem angegebenen Pfad entspricht. |
| GetDriveName | Gibt den Laufwerksnamen eines angegebenen Pfads zurück. |
| GetExtensionName | Gibt den Dateierweiterungsnamen der letzten Komponente in einem angegebenen Pfad zurück. |
| GetFile | Gibt ein Dateiobjekt für einen angegebenen Pfad zurück. |
| GetFileName | Gibt den Dateinamen oder den Ordnernamen für die letzte Komponente in einem angegebenen Pfad zurück. |
| GetFolder | Gibt ein Ordnerobjekt für einen angegebenen Pfad zurück. |
| GetParentFolderName | Gibt den Namen des übergeordneten Ordners der letzten Komponente in einem angegebenen Pfad zurück. |
| GetSpecialFolder | Gibt den Pfad zu einigen speziellen Ordnern von Windows zurück. |
| GetTempName | Gibt eine nach dem Zufallsprinzip generierte temporäre Dateien oder einen temporären Ordner zurück. |
| Move | Verschiebt eine bestimmte Datei oder einen Ordner an einem Speicherort an einen anderen. |
| MoveFile | Verschiebt eine oder mehrere Dateien von einem Speicherort an einen anderen. |

| Methode | Beschreibung |
|------------------|--|
| MoveFolder | Verschiebt einen oder mehrere Ordner von einem Speicherort an einen anderen. |
| OpenAsTextStream | Öffnet eine angegebene Datei und gibt ein TextStream-Objekt zurück, das zum Lesen aus, zum Schreiben in oder zum Anfügen an die Datei verwendet werden kann. |
| OpenTextFile | Öffnet eine Datei und gibt ein TextStream-Objekt zurück, das zum Zugreifen auf die Datei verwendet werden kann. |
| WriteLine | Schreibt eine angegebene Zeichenfolge und Zeichen für eine neue Zeile in eine TextStream-Datei. |

Die Eingabe eines Verzeichnisses muss auf abschließende Schrägstriche \ (Backslash) geprüft werden, bevor sie als Dateipfad verwendet werden kann. Die Methode BuildPath vereinfacht die Nutzung.

Codeliste 5. Die Prozedur zeigt die Anwendung der Methode BuildPath

```
Sub TestBuildPath()
    Const sSourceFilePath As String = "C:\Temp" 'ohne Backslash
    Const sTargetFilePath As String = "C:\Temp\" 'mit Backslash
    Const sFileName As String = "Text.txt"

    Dim fsoTemp As FileSystemObject
    Set fsoTemp = CreateObject("Scripting.FileSystemObject")

    Debug.Print fsoTemp.BuildPath(sSourceFilePath, sFileName)
    Debug.Print fsoTemp.BuildPath(sTargetFilePath, sFileName)
    Set fsoTemp = Nothing
End Sub
```

Die Ausgabe ins Direktfenster zeigt den gleichen Text:

```
C:\Temp\Test.txt
C:\Temp\Test.txt.
```

Die Methode *CopyFile* kopiert eine oder mehrere Dateien von einem Ordner in einen anderen.

*Codeliste 6. Die Prozedur kopiert alle Textdateien *.txt aus dem Ordner C:\Temp zum Ordner C:\Neu*

```
Sub TestCopyFile()
    Dim fsoTemp As FileSystemObject

    Set fsoTemp = CreateObject("Scripting.FileSystemObject")
    fsoTemp.CopyFile "C:\Temp\*.txt", "C:\Neu\"
    Set fsoTemp = Nothing
End Sub
```

Platzhalterzeichen können nur in der letzten Pfadkomponente des Quellverzeichnisses verwendet werden. Das gilt auch für die folgende Methode.

Analog kopiert die Methode *CopyFolder* einen Ordner rekursiv an einen anderen Speicherort.

Codeliste 7. Die Prozedur kopiert den Inhalt von Ordner C:\Temp in den Ordner C:\Neu

```
Sub TestCopyFolder()  
    Dim fsoTemp As FileSystemObject  
  
    Set fsoTemp = CreateObject("Scripting.FileSystemObject")  
    fsoTemp.CopyFolder "C:\Temp", "C:\Neu"  
    Set fsoTemp = Nothing  
End Sub
```

Die Methode *CreateFolder* erzeugt einen neuen Ordner.

Codeliste 8. Die Prozedur kopiert den Inhalt von Ordner C:\Temp in den Ordner C:\Neu

```
Sub TestCreateFolder()  
    Dim fsoTemp As FileSystemObject  
  
    Set fsoTemp = CreateObject("Scripting.FileSystemObject")  
    fsoTemp.CreateFolder "C:\Neu"  
    Set fsoTemp = Nothing  
End Sub
```

Die Methode *FolderExists* prüft die Existenz des angegebenen Ordners und liefert als Ergebnis einen der booleschen Werte *True* (Ordner existiert) oder *False* (Ordner existiert nicht).

Codeliste 9. Die Prozedur prüft die Existenz eines Ordners

```
Sub TestFolderExists()  
    Dim fsoTemp As FileSystemObject  
    Dim vReturn As Variant  
  
    Set fsoTemp = CreateObject("Scripting.FileSystemObject")  
    vReturn = fsoTemp.FolderExists("C:\Neu")  
    Set fsoTemp = Nothing  
End Sub
```

Analog erfolgt die Existenzprüfung eines Files.

Codeliste 10. Die Prozedur prüft die Existenz eines Files

```
Sub TestFileExists()  
    Dim fsoTemp As FileSystemObject  
    Dim vReturn As Variant  
  
    Set fsoTemp = CreateObject("Scripting.FileSystemObject")  
    vReturn = fsoTemp.FileExists("C:\Temp\Test.txt")  
    Set fsoTemp = Nothing  
End Sub
```

Die Prozedur *GetTempName* gibt einen willkürlich generierten Namen einer temporären Datei oder eines temporären Ordners zurück, der bei Vorgängen nützlich ist, bei denen eine temporäre Datei oder ein temporärer Ordner benötigt wird.

Codeliste 11. Die Prozedur erstellt eine temporäre Textdatei und schreibt darin einen Text

```
Sub TestGetTempName()  
    Dim fsoTemp As FileSystemObject  
    Dim sFileName As String  
    Dim tsmTemp As TextStream  
  
    Set fsoTemp = CreateObject("Scripting.FileSystemObject")
```

```

sFileName = fsoTemp.GetTempName()
Debug.Print sFileName
'Textdatei erstellen
Set tsmTemp = fsoTemp.CreateTextFile(sFileName)
tsmTemp.WriteLine "Dies ist ein Test!"
tsmTemp.Close
Set tsmTemp = Nothing
Set fsoTemp = Nothing
End Sub

```

Die Prozedur liefert in diesem Fall den temporären Dateinamen rad77184.tmp im Direktfenster, der natürlich jedesmal anders ist.

Das folgende Beispiel zeigt die Anwendung der Methode *OpenTextFile* zum Anhängen von Text.

Codeliste 12. Die Prozedur hängt Text an den Inhalt einer bestehenden Textdatei an

```

Sub TestOpenTextFile()
Dim fsoTemp As FileSystemObject
Dim sFileName As String
Dim tsmTemp As TextStream

Set fsoTemp = CreateObject("Scripting.FileSystemObject")
Set tsmTemp = fsoTemp.OpenTextFile _
("C:\Temp\Test.txt", _
ForAppending, _
True, _
TristateFalse)
tsmTemp.WriteLine "Dies ist ein Test!"
tsmTemp.Close
Set tsmTemp = Nothing
Set fsoTemp = Nothing
End Sub

```

Neben *ForAppending* zum Anhängen von Text, öffnet eine Textdatei auch mit *ForReading* nur zum Lesen und *ForWriting* nur zum Schreiben. Der boolesche Wert gibt an, ob eine neue Datei erstellt werden darf (True) oder nicht (False). Das letzte Argument gibt das Format der Datei an, mit *TristateUseDefault* nach Systemvorgabe, *TristateTrue* als Unicode- und *TristateFalse* als Ascii-Datei.

3 Eigenschaften

Tabelle 2. Eigenschaften des FileSystemObject-Objekts

| Eigenschaft | Beschreibung |
|-------------|--|
| Drives | Gibt eine Auflistung aller Laufwerksobjekte auf dem Computer zurück. |
| Name | Legt den Namen einer angegebenen Datei oder eines angegebenen Ordners fest oder gibt diesen zurück. |
| Path | Gibt den Pfad für eine angegebene Datei, einen angegebenen Ordner oder ein angegebenes Laufwerk zurück. |
| Size | Für Dateien wird die Größe der angegebenen Datei in Byte zurückgegeben; für Ordner wird die Größe aller Dateien und Unterordner in Byte zurückgegeben, die in dem Ordner enthalten sind. |

| | |
|------|---|
| Type | Gibt Informationen über den Typ einer Datei oder eines Ordners zurück (für Dateien mit der Endung .TXT wird beispielsweise „Textdokument“ zurückgegeben). |
|------|---|

Die Eigenschaft *Drives* gibt eine Auflistung zurück, die alle Drive-Objekte enthält, die auf dem lokalen Computer verfügbar sind. In Laufwerken für Wechseldatenträger müssen keine Datenträger eingelegt sein, damit sie in der Auflistung aufgeführt werden.

Codeliste 13. Die Prozedur liefert eine Auflistung aller verfügbaren Laufwerke

```

Sub TestExistsDrive()
    Dim fsoTemp As FileSystemObject
    Dim drsTemp As Drives
    Dim drvTemp As Drive
    Dim sText As String
    Dim sName As String

    Set fsoTemp = CreateObject("Scripting.FileSystemObject")
    Set drsTemp = fsoTemp.Drives
    sText = ""
    On Error Resume Next
    For Each drvTemp In drsTemp
        sText = sText & drvTemp.DriveLetter & ": "
        If drvTemp.DriveType = 3 Then
            sName = drvTemp.ShareName
        Else
            sName = drvTemp.VolumeName
        End If
        sText = sText & sName & vbCrLf
    Next
    MsgBox sText
    Set fsoTemp = Nothing
    Set drsTemp = Nothing
End Sub

```

Die Laufwerkskennungen A: und B: stammen aus den Anfängen des PCs und sind für zwei Diskettenlaufwerke reserviert. Nachfolgend ist ein Ergebnisbeispiel dargestellt (Bild 2).

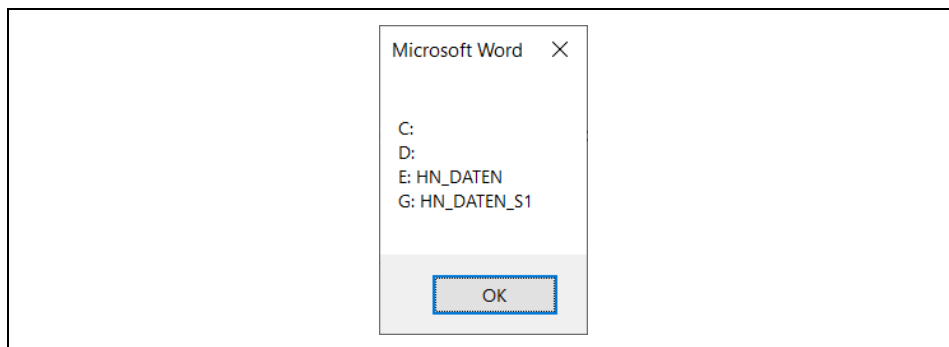


Bild 2. Beispielausgabe einer Laufwerksliste

Die Eigenschaften *Name*, *Path*, *Size* und *Type* liefern die Zugriffsdaten einer Datei.

Codeliste 14. Die Prozedur zeigt die Zugriffsdaten einer vorhandenen Datei

```
Sub ShowFileAccessInfo()  
    Dim fsoTemp As FileSystemObject  
    Dim filTemp As File  
    Dim sFileName As String  
    Dim sText As String  
  
    Set fsoTemp = CreateObject("Scripting.FileSystemObject")  
    Set filTemp = fsoTemp.GetFile("C:\Temp\Test.txt")  
    With filTemp  
        sText = .Name & " auf Laufwerk " & UCase(.Drive) & vbCrLf  
        sText = sText & "Verzeichnis: " & UCase(.Path) & vbCrLf  
        sText = sText & "Größe: " & .Size & " Bytes" & vbCrLf  
        sText = sText & "Type: " & .Type & vbCrLf  
        sText = sText & "Erstellt: " & .DateCreated & vbCrLf  
        sText = sText & "Letzter Zugriff: " & .DateLastAccessed & vbCrLf  
        sText = sText & "Letzte Änderung: " & .DateLastModified  
    End With  
    MsgBox sText, 0, "Datei-Info"  
    Set fsoTemp = Nothing  
    Set filTemp = Nothing  
End Sub
```

Die Daten werden in einem Ausgabefenster gezeigt (Bild 3).

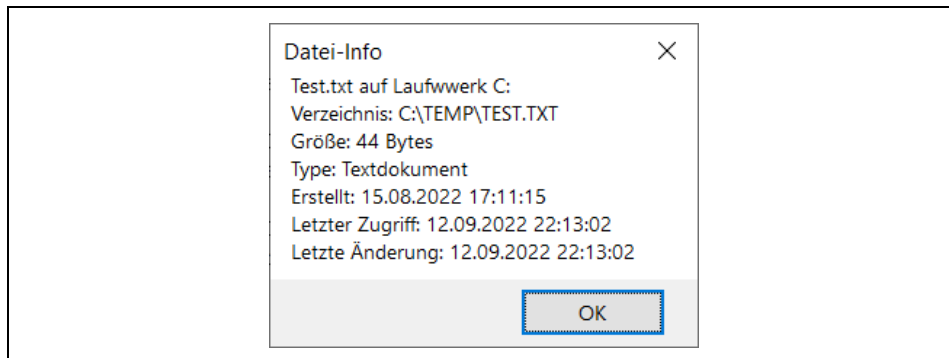


Bild 3. Beispiel für Zugriffsdaten einer Datei