

# Dateien

## Inhaltsverzeichnis

- [1 Allgemeine Attribute](#)
- [2 FileCopy](#)
- [3 FileMove](#)
- [4 FileRename](#)
- [5 FileDelete](#)
- [6 FileExists](#)
- [7 FileCreateSymbolicLink](#)
- [8 FileGetCreationTime](#)
- [9 FileSetCreationTime](#)
- [10 FileIsReadOnly](#)
- [11 FileSetReadOnly](#)
- [12 FileSetWriteable](#)
- [13 FileVersion](#)
- [14 GetFileName](#)
- [15 GetFileNameWithoutExtension](#)
- [16 GetExtension](#)
- [17 FileListDirectory](#)
- [18 CreateShortcut](#)
- [19 Zip](#)
- [20 Unzip](#)
- [21 FileMirror](#)

Die Aktionen aus der Kategorie Dateien

# 1 Allgemeine Attribute

Die allgemeinen Attribute **IgnoreError**, **Variable** und **Condition** können bei allen Aktionen angegeben werden. Die Attribute sind optional und brauchen nur bei Bedarf hinterlegt werden. Wenn diese für eine Aktion nicht benötigt werden, können diese aber auch zur besseren Lesbarkeit des Skriptes entfernt werden.

## IgnoreError

Das optionale Attribut **IgnoreError** gibt an, ob bei einem Fehler die Ausführung des Batchpad Skriptes abbricht oder das Skript weiter ausgeführt werden soll. Der Wert muss dem Typ Boolean (true oder false) entsprechen.

## Variable

Das optionale Attribut **Variable** kann immer dann verwendet werden, wenn man das Ergebnis einer auszuführenden Aktion ermitteln möchte. Variable="{@ResultFileExists}".

Die Ergebnisse sind je nach ausgeführter Aktion vom Typ her unterschiedlich, oft ist es ein Boolean (true oder false) der angibt ob die Aktion erfolgreich war. Bei Aktionen für Zeichenketten sind die Ergebnisse dann eher vom Typ String usw.

## Condition

Das optionale Attribut **Condition** gibt an, ob die Aktion ausgeführt werden soll. Hierzu wird der Inhalt des Attributes als logischer Ausdruck auf Wahr oder Falsch geprüft. Der Ausdruck sollte dem Typ Boolean (true oder false) entsprechen.

Der Ausdruck kann Funktionen aus [VBScript](#) enthalten, genauso wie Operatoren NOT, OR, AND...

Mit dem Condition Attribut wertet man in der Regel Variablen aus, die Ergebnisse aus zuvor durchgeführten Aktionen enthalten. Beispiel: Condition="NOT {@ResultFileExists}"

# 2 FileCopy

Die Aktion **FileCopy** kopiert eine angegebene Datei (Attribut: *Source*) zu dem angegebenen Ziel (Attribut: *Destination*). Mit dem Attribut *Overwrite* kann bestimmt werden, ob eine bereits existierende Datei überschrieben werden soll.

```
<FileCopy Source="" Destination="" Overwrite="true" Condition="" Variable="{@Result}" IgnoreError="false" />
```

# 3 FileMove

Die Aktion **FileMove** verschiebt eine angegebene Datei (Attribut: *Source*) in das angegebene Ziel (Attribut: *Destination*).

```
<FileMove Source="" Destination="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

# 4 FileRename

Die Aktion **FileRename** verändert den Pfad, Dateinamen und Dateieindung einer angegebenen Datei (Attribut: *Source*) und weist der Datei einen neuen Pfad, Dateinamen und Dateieindung zu (Attribut: *Destination*).

```
<FileRename Source="" Destination="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 5 FileDelete

Die Aktion **FileDelete** löscht eine angegebene Datei (Attribut: *File*).

```
<FileDelete File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 6 FileExists

Die Aktion **FileExists** prüft, ob eine angegebene Datei (Attribut: *File*) existiert. Falls Ja wird in das Attribut *Variable* true zurückgegeben

```
<FileExists File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 7 FileCreateSymbolicLink

Die Aktion **FileCreateSymbolicLink** erstellt eine symbolische Verknüpfung zu einer Datei innerhalb des Dateisystems.

```
<FileCreateSymbolicLink Source="" Destination="" Variable="{@Result}" />
```

## 8 FileGetCreationTime

Die Aktion **FileGetCreationTime** ermittelt zu einer angegebenen Datei (Attribut: *File*) das Erstellungsdatum liefert das Ergebnis in das Attribut *Variable* zurück.

```
<FileGetCreationTime File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 9 FileSetCreationTime

Mit der Aktion **FileSetCreationTime** versieht das Erstellungsdatum einer angegebenen Datei (Attribut: *File*) mit einem neuen Zeitstempel (Attribut: *Value*).

```
<FileSetCreationTime File="" Value="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 10 FileIsReadOnly

Die Aktion **FileIsReadOnly** prüft, ob die angegebene Datei (Attribut: *File*) schreibgeschützt ist.

```
<FileIsReadOnly File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 11 FileSetReadOnly

Die Aktion **FileSetReadOnly** versieht die angegebene Datei (Attribut: *File*) mit einem Schreibschutz.

```
<FileSetReadOnly File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 12 FileSetWritable

Die Aktion **FileSetWriteable** entfernt den Schreibschutz der angegebenen Datei (Attribut: *File*).

```
<FileSetWriteable File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 13 FileVersion

Die Aktion **FileVersion** ermittelt zu einer angegebenen Datei (Attribut: *File*) die Dateiversion. Ist das Attribut *ProductVersion* auf *True* gesetzt, wird die Produktversion der Datei in das Attribut *Variable* zurückgegeben.

```
<FileVersion File="" ProductVersion="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 14 GetFileName

Die Aktion **GetFileName** gibt den Dateinamen inkl. der Dateiendung der angegebenen Datei (Attribut: *Source*) in das Attribut *Variable* zurück.

```
<GetFileName Source="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 15 GetFileNameWithoutExtension

Die Aktion **GetFileNameWithoutExtension** gibt den Dateinamen ohne die Dateiendung der angegebenen Datei (Attribut: *Source*) in das Attribut *Variable* zurück.

```
<GetFileNameWithoutExtension Source="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 16 GetExtension

Die Aktion **GetExtension** gibt die Dateiendung der angegebenen Datei (Attribut: *Source*) in das Attribut *Variable* zurück. Hinweis: Die Dateiendung wird mit dem Trennzeichen "." übernommen z.B. ".pdf".

```
<GetExtension Source="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 17 FileListDirectory

Die Aktion **FileListDirectory** erstellt ein DataTable Objekt, das Dateien und Ordner enthält, die unter dem angegebenen Pfad (Attribut: *Source*) verfügbar sind. Dabei kann angegeben werden ob nur Dateien (Attribut: *OnlyFiles*) oder nur Ordner (Attribut: *OnlyDirectories*) in das DataTable Objekt übernommen werden. Zusätzliche Informationen können über das Attribut *WithInformations* angefordert werden. Die Informationen, die ein DataTable Objekt zur Laufzeit hält, können über den Inspektor eingesehen werden (Hauptartikel: Inspektor & Debugging). Dieses Objekt kann dazu verwendet werden, um in einer ForEach Schleife über die Einträge des DataTable Objekts zu iterieren. Siehe Beispiel: [Datensicherung](#).

```
<FileListDirectory Source="" Data="{@myData}" OnlyFiles="false" OnlyDirectories="false" WithInformations="false" DataCount="{@ResultCount}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 18 CreateShortcut

Die Aktion **CreateShortcut** erstellt eine Dateiverknüpfung zu der angegebenen Datei (Attribut: *Source*) unter einem angegebenen Zielpfad (Attribut: *Destination*). Der Name und das Icon der Dateiverknüpfung werden über die gleichnamigen Attribute gesetzt. Argumente, die dem Aufruf des Shortcuts angehängt werden sollen, werden über das Attribut *Arguments* angegeben. Über das Attribut *Workpath* kann angegeben werden in welchem Verzeichnis die Dateiverknüpfung ausgeführt wird.

```
<CreateShortcut Source="" Destination="" Name="" Icon="" Workpath="" Arguments="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 19 Zip

Die Aktion **Zip** packt eine angegebene Datei/Verzeichnis (Attribut: *Source*) in ein Zip-Archiv (Attribut: *Destination*).

```
<Zip Source="" Destination="" Password="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 20 Unzip

Die Aktion **UnZip** entpackt ein angegebenes Zip-Archiv (Attribut: *Source*) in ein Zielverzeichnis (Attribut: *Destination*).

```
<UnZip Source="" Destination="" Password="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 21 FileMirror

Die Aktion **FileMirror** kopiert Dateien von einer Quelle (Attribut: *Source*) in ein Zielverzeichnis (Attribut: *Destination*). Über das Attribut *Message* wird ein Bericht des Kopiervorgangs zurückgegeben. Verzeichnisse und Dateien, die ausgeschlossen werden sollen, können über die Attribute *ExcludeFiles* und *ExcludeDirectories* angegeben werden. Mehrere Verzeichnisse bzw. Dateien werden dabei mit einem Semikolon getrennt. Zudem kann eine Wildcard (\*) verwendet werden (siehe Beispiel).

```
<FileMirror Source="" Destination="" ExcludeFiles="" ExcludeDirectories="" Message="{@ResultMessage}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

Beispiel:

Code

```
<Batch>  
@Echo off  
@If Exist "%Source%" (echo "%Source%" is not a file) else  
<PrintText="{@ResultMessage}" Condition="Variable="{@Result}" IgnoreError="false">  
</Batch>
```