

Berechnungen

Inhaltsverzeichnis

- [1 Allgemeine Attribute](#)
- [2 CalculateAddition](#)
- [3 CalculateSubtraction](#)
- [4 CalculateMultiplication](#)
- [5 CalculateDivision](#)
- [6 CalculateMax](#)
- [7 CalculateMin](#)
- [8 ConvertToNumber](#)
- [9 MathAbs](#)
- [10 MathTruncate](#)
- [11 MathRound](#)
- [12 MathFloor](#)
- [13 MathCeiling](#)

Die Aktionen aus der Kategorie Berechnungen

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 CalculateAddition

Die Aktion **CalculateAddition** führt eine Addition mit den Werten in den Attributen *Value1* und *Value2* aus und gibt das Ergebnis über das Attribute *Variable* aus. Über das Attribut *Float* kann angegeben werden, ob eine Fließkommazahl zurückgegeben werden soll.

```
<CalculateAddition Value1="" Value2="" Float="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

3 CalculateSubtraction

Die Aktion **CalculateSubtraction** führt eine Subtraktion mit den Werten in den Attributen *Value1* und *Value2* aus und gibt das Ergebnis über das Attribute *Variable* aus. Dabei wird *Value2* von *Value1* subtrahiert. Über das Attribut *Float* kann angegeben werden, ob eine Fließkommazahl zurückgegeben werden soll.

```
<CalculateSubtraction Value1="" Value2="" Float="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

4 CalculateMultiplication

Die Aktion **CalculateMultiplication** führt eine Multiplikation mit den Werten in den Attributen *Value1* und *Value2* aus und gibt das Ergebnis über das Attribute *Variable* aus. Über das Attribut *Float* kann angegeben werden, ob eine Fließkommazahl zurückgegeben werden soll.

```
<CalculateMultiplication Value1="" Value2="" Float="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

5 CalculateDivision

Die Aktion **CalculateDivision** führt eine Subtraktion mit den Werten in den Attributen *Value1* und *Value2* aus und gibt das Ergebnis über das Attribute *Variable* aus. Dabei wird *Value1* durch *Value2* dividiert. Über das Attribut *Float* kann angegeben werden, ob eine Fließkommazahl zurückgegeben werden soll.

```
<CalculateDivision Value1="" Value2="" Float="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

6 CalculateMax

Die Aktion **CalculateMax** bestimmt den maximalen Wert der Attribute *Value1* und *Value2* und gibt den maximalen Wert der über das Attribute *Variable* aus.

```
<CalculateMax Value1="" Value2="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

7 CalculateMin

Die Aktion **CalculateMin** bestimmt den minimalen Wert der Attribute *Value1* und *Value2* und gibt den minimalen Wert der über das Attribute *Variable* aus.

```
<CalculateMin Value1="" Value2="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

8 ConvertToNumber

Die Aktion **ConvertToNumber** konvertiert eine Zeichenfolge (Attribut: *Value*) und gibt eine ganze Zahl oder Fließkommazahl zurück (Attribut: *Variable*). Über das Attribut *Float* kann angegeben werden, ob eine Fließkommazahl zurückgegeben werden soll.

```
<ConvertToNumber Value="" Float="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

9 MathAbs

Die Aktion **MathAbs** gibt den absoluten Wert einer Zahl (Attribut: *Value*) in das Attribut *Variable* zurück. Über das Attribut *Float* kann angegeben werden, ob der Wert in dem Attribut *Value* eine Fließkommazahl enthält.

```
<MathAbs Value="" Float="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

10 MathTruncate

Die Aktion **MathTruncate** berechnet den ganzzahligen Teil der Dezimalzahl, die in dem Attribut *Value* hinterlegt ist und gibt den Wert über das Attribut *Variable* zurück.

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

11 MathRound

Die Aktion **MathRound** rundet einen Dezimalwert (Attribut: *Value*) auf den nächstgelegenen ganzzahligen Wert und Mittelpunktwerte auf die nächstgelegene gerade Zahl und gibt den berechneten Wert über das Attribut *Variable* zurück.

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

12 MathFloor

Die Aktion **MathFloor** gibt die größte Ganzzahl zurück (Attribut: *Variable*), die kleiner oder gleich der angegebenen Zahl (Attribut: *Value*) ist.

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

13 MathCeiling

Die Aktion **MathCeiling** gibt die kleinste Ganzzahl zurück (Attribut: *Variable*), die größer oder gleich der angegebenen Zahl (Attribut: *Value*) ist.

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