Zeichenketten

Inhaltsverzeichnis

- 1 Allgemeine Attribute
- 2 StringLeft
- 3 StringMid
- 4 StringRight
- 5 StringLength
- 6 StringReplace
- 7 StringContains
- 8 StringIndexOf
- 9 StringStartsWith
- 10 StringEndsWith
- 11 StringToLower
- 12 StringToUpper
- 13 StringTrim
- 14 StringFormat
- 15 StringAppend
- 16 StringEmpty
- 17 StringSplit
- 18 StringBetween

Die Aktionen aus der Kategorie Zeichenketten

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 StringLeft

Die Aktion **StringLeft** gibt eine Zeichenfolge zurück (Attribut: *Variable*), abhängig von der eingestellten Länge (Attribut: *Length*), von links nach rechts, der in dem Attribut *Value* hinterlegten Zeichenfolge.

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

3 StringMid

Die Aktion **StringMid** gibt eine Zeichenfolge zurück (Attribut: *Variable*), die eine festgelegte Zeichenanzahl (Attribut: *Length*) aus einer Zeichenfolge (Attribut: *Value*) enthält. Über das Attribut *Start* kann von links nach rechts die Startposition als ganze Zahl angegeben werden.

```
<StringMid Value="" Start="" Length="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

4 StringRight

Die Aktion **StringRight** gibt eine Zeichenfolge zurück (Attribut: *Variable*), mit einer bestimmten Anzahl von Zeichen (Attribut: *Length*) ab dem ersten Zeichen von rechts einer Zeichenfolge (Attribut: *Value*).

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

5 StringLength

Die Aktion **StringLength** gibt eine ganze Zahl (Attribut: *Variable*) zurück, in der die Anzahl der Zeichen in einer Zeichenfolge (Attribute: *Value*) enthalten ist.

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

6 StringReplace

Die Aktion **StringReplace** gibt eine Zeichenfolge zurück (Attribut: *Variable*), in der eine bestimmte untergeordnete Zeichenfolge (Attribut: *Search*) durch eine Zeichenfolge (Attribut: *Replace*) ersetzt wurde.

```
<StringReplace Value="" Search="" Replace="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

7 StringContains

Die Aktion **StringContains** gibt true oder false zurück (Attribut: *Variable*), ob eine Zeichenfolge (Attribute: *Search*) in einer weiteren Zeichenfolge (Attribute: *Value*) enthalten ist. Über das Attribut *Not* kann das Ergebnis negiert werden.

```
<StringContains Value="" Search="" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

8 StringIndexOf

Die Aktion **StringIndexOf** gibt eine ganze Zahl zurück (Attribut: *Variable*), in der die erste Position einer Zeichenfolge (Attribute: *Search*) in einer Zeichenfolge (Attribute: *Value*) von links nach rechts auftritt.

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

9 StringStartsWith

Die Aktion **StringStartsWith** gibt true oder false zurück (Attribut: *Variable*), ob eine Zeichenfolge (Attribute: *Value*) mit einer bestimmten Zeichenfolge (Attribute: *Search*) beginnt. Über das Attribut *Not* kann das Ergebnis negiert werden.

```
<StringStartsWith Value="" Search="" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

10 StringEndsWith

Die Aktion **StringEndsWith** gibt true oder false zurück (Attribut: *Variable*), ob eine Zeichenfolge (Attribute: *Value*) mit einer bestimmten Zeichenfolge (Attribute: *Search*) endet. Über das Attribut *Not* kann das Ergebnis negiert werden.

```
<StringEndsWith Value="" Search="" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

11 StringToLower

Die Aktion **StringToLower** gibt eine Zeichenfolge oder ein Zeichen zurück (Attribut: *Variable*), die bzw. das die angegebene Zeichenfolge (Attribute: *Value*) in Kleinbuchstaben enthält.

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

12 StringToUpper

Die Aktion **StringToUpper** gibt eine Zeichenfolge oder ein Zeichen zurück (Attribut: *Variable*), die bzw. das die angegebene Zeichenfolge (Attribute: *Value*) in Großbuchstaben enthält.

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

13 StringTrim

Die Aktion **StringTrim** gibt eine Zeichenfolge zurück (Attribut: *Variable*), die eine Kopie einer angegebenen Zeichenfolge (Attribute: *Value*) ohne führende oder nachgestellte Leerzeichen enthält.

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

14 StringFormat

Die Aktion **StringFormat** gibt eine Zeichenfolge zurück (Attribut: *Variable*), die entsprechend den Anweisungen in einem Formatausdruck (Attribut: *Format*) formatiert ist.

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

15 StringAppend

Die Aktion **StringAppend** gibt eine Zeichenfolge zurück (Attribut: *Variable*), die durch Verknüpfen mehrerer Teilzeichenfolgen (Attribute: *Value1* und *Value2*) erstellt wurde. Über das Attribut *NewLine* kann festgelegt werden, ob die Teilzeichenfolgen mit einem Zeilenumbruch verbunden werden sollen.

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

16 StringEmpty

Die Aktion **StringEmpty** prüft, ob die Zeichenfolge in dem Attribut *Value* leer ist und gibt bei einem leeren String true zurück, ansonsten false. Über das Attribut *Not* kann das Ergebnis negiert werden.

```
<StringEmpty Value="" Not="false" Condition="" Variable="{@Result}" IgnoreError="false" />
```

17 StringSplit

Die Aktion **StringSplit** gibt ein eindimensionales Array zurück (Attribut: *Data*), dass eine Anzahl von Teilzeichenfolgen enthält. Die Teilzeichenfolgen werden dabei entweder von einem angegebenen Trennzeichen (Attribut: *Delimiter*) bestimmt, alternativ von einem Zeilenumbruch (Attribut: *SplitLines*). Über das Attribut *RemoveEmptyValues* wird bestimmt ob leere Einträge übernommen (false) oder verworfen werden (true). Über das Data-Objekt kann anschließend über eine ForEach Schleife iteriert werden (Siehe:

Allgemein/ForEach).

```
<StringSplit Data="{@myData}" Value="" Delimiter=";" SplitLines="false" RemoveEmptyValues="false" DataCount="{@ResultCount}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

18 StringBetween

Die Aktion **StringBetween** gibt eine Zeichenfolge zurück (Attribut: *Variable*), welche zwischen zwei Zeichenfolgen (Attribute: StartText und EndText) einer bestimmten Zeichenfolge liegt (Attribut: *Value*).

```
<StringBetween Value="" StartText="" EndText="" Condition="" Variable="{@Result}" IgnoreError="false" />
```