CSV & Text

Inhaltsverzeichnis

Die Aktionen aus der Kategorie "CSV & Text"

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 CSVReadData

Die Aktion **CSVReadData** gibt ein DataTable Objekt zurück (Attribut: *Data*), welches aus einer CSV Datei ausgelesen wird (Attribut: *File*). Das Attribut *Delimiter* bestimmt mit welchem Zeichen die einzulesenden Einträge getrennt sind. Über das Attribut *FirstLineIsHeader* wird angegeben, dass die erste Reihe für die Bezeichnung der Spalten herangezogen wird. Über diese Bezeichnung werden die Spalten beschriftet und können über diese Bezeichnung über das DataTable Objekt angesprochen werden (Siehe Allgemein/ForEach). Werden keine Spaltenbezeichnungen angegeben, werden die Spalten aufsteigend mit ganzen Zahlen beginnend bei 0 beschriftet. Die Codierung der Datei wird über das Attribut *Codepage* angegeben. Sind die Felder oder einzelne Felder mit Anführungszeichen versehen, und die Felder sollen ohne zusätzliche Anführungszeichen in das DataTable Objekt übernommen werden, muss das Attribut *HasFieldsEnclosedInQuotes* auf true gesetzt werden. Über das Attribut *DataCount* wird die Anzahl der Datensätze zurückgegeben.

```
<CSVReadData Data="{@myData}" File="" Codepage="1252" Delimiter=";" HasFieldsEnclosedInQuotes
="true" FirstLineIsHeader="true" DataCount="{@ResultCount}" Condition="" Variable="{@Result}"
IgnoreError="false" />
```

3 CSVWriteData

Die Aktion **CSVWriteData** schreibt ein DataTable Objekt (Attribut: *Data*) in eine CSV Datei (Attribut: *File*). Das Attribut *Delimiter* bestimmt mit welchem Zeichen die zu schreibenden Zeichen getrennt werden. Über das Attribut *FirstLineIsHeader* wird angegeben, dass die erste Reihe für die Bezeichnung der Spalten herangezogen wird. Werden keine Spaltenbezeichnungen angegeben, wird die CSV Datei ohne einen Kopfbereich erstellt. Die Codierung der Datei wird über das Attribut *Codepage* angegeben. Sind die Felder oder einzelne Felder mit Anführungszeichen versehen, und die Felder sollen mit dem Anführungszeichen in die CSV Datei übernommen werden, muss das Attribut *HasFieldsEnclosedInQuotes* auf true gesetzt werden.

```
<CSVWriteData Data="{@myData}" File="" Codepage="1252" Delimiter=";" HasFieldsEnclosedInQuotes
="true" FirstLineIsHeader="true" Condition="" Variable="{@Result}" IgnoreError="false" />
```

4 TextFileRead

Die Aktion **TextFileRead** gibt einen Text zurück (Attribut *Variable*). Über das Attribut *File* wird die auszulesende Datei angegeben. Die Codierung der Datei wird über das Attribut *Codepage* angegeben.

```
<TextFileRead File="" Codepage="1252" Condition="" Variable="{@Result}" IgnoreError="false" />
```

5 TextFileWrite

Die Aktion **TextFileWrite** schreibt einen bestimmten Text (Attribut *Content*) eine Datei (Attribut: *File*). Die Codierung der Datei wird über das Attribut *Codepage* angegeben. Soll der Text an einen Text in einer bestehenden Textdatei angehangen werden, muss das Attribut *Append* auf true gesetzt werden. Der Text wird dabei ohne einen Zeilenumbruch an den bestehenden Text angehangen.

```
<TextFileWrite File="" Content="" Append="false" Codepage="1252" Condition="" Variable="{@Result}"
IgnoreError="false" />
```

6 ExcelReadData

```
<ExcelReadData Data="{@myData}" File="" WorksheetName="" StartAtLine="0" StartLineIsHeader="false"
Variable="{@Result}" />
```

7 TextConvertCodepage

```
<TextConvertCodepage Text="" FromCodepage="1252" ToCodepage="65001" Variable="{@Result}" />
```

8 TextClipboardRead

```
<TextClipboardRead Variable="{@Result}" />
```

9 TextClipboardWrite

```
<TextClipboardWrite Content="" Variable="{@Result}" />
```