Datenbank MS-SQL

Inhaltsverzeichnis

Die Aktionen aus der Kategorie "MS-SQL Datenbank"

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 SQLConnect

Die Aktion **SQLConnect** stellt mit den angegebenen Parametern eine Verbindung zur Datenbank her. Mit dem Attribut *Server* wird der Servername angeben, mit *Database* die Datenbank, an die sich angemeldet werden soll. Das Attribut *User* für den Benutzernamen und *Password* für das Passwort dienen zur Authentifizierung am Datenbankserver. Alternativ zu diesen Angaben kann über das Attribut *ConnectionString* auch ein kompletter ConnectionString verwendet werden.

```
<SQLConnect Connection="{@myConnection}" Server="" Database="" User="" Password="" Port="" ConnectionString="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

3 SQLExecute

Die Aktion **SQLExecute** sendet eine SQL-Anweisung an die Datenbank und liefert bei erfolgreicher Ausführung *True* als Wert in die angegebene *Variable* zurück. Mit dem Attribut *Query* wird die auszuführende SQL-Anweisung angegeben.

```
<SQLExecute Query="" Connection="{@myConnection}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

4 SQLExecuteIdentity

Die Aktion **SQLExecuteIdentity** gibt die ID der ausgeführten INSERT SQL-Anweisung (Attribut: Query) über das Attribut *Variable* zurück.

```
<SQLExecuteIdentity Query="" Connection="{@myConnection}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

5 SQLExists

Die Aktion **SQLExists** prüft ob in einer Tabelle abhängig von einem Filterkriterium Datensätze vorhanden sind. Mit dem Attribut *Table* wird der Name der Tabelle übergeben, die überprüft werden soll. Über das Attribut *Where* wird das Filterkriterium festgelegt.

```
<SQLExists Table="" Where="" Connection="{@myConnection}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

6 SQLExistsField

Die Aktion **SQLExistsField** prüft ob in einer Tabelle eine bestimmte Spalte vorhanden ist. Mit dem Attribut *Table* wird der Name der Tabelle übergeben, in der nach der Spalte gesucht werden soll. Über das Attribut *Field* wird die zu suchende Spalte festgelegt.

```
<SQLExistsField Table="" Field="" Connection="{@myConnection}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

7 SQLExistsTable

Die Aktion **SQLExistsTable** prüft ob eine Tabelle in der angegeben Datenverbindung *Connection* vorhanden ist. Mit dem Attribut *Table* wird der Name der Tabelle übergeben.

```
<SQLExistsTable Table="" Connection="{@myConnection}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

8 SQLLookup

Die Aktion **SQLLookup** liest einen einzelnen Wert aus der Datenbank. Mit dem Attribut *Table* wird die Tabelle angegeben und mit *Field* der Feldnamen dessen Wert ausgelesen werden soll. Über das Attribut *Where* wird der Filter auf den auszulesenden Datensatz bestimmt. Mit dem Attribut *Variable* wird die Platzhalter-Variable bestimmt, in der sich nach Ausführung der Aktion der ausgelesene Wert befindet.

```
<SQLLookup Table="" Field="" Where="" Connection="{@myConnection}" DefaultValue="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

9 SQLReadData

Die Aktion **SQLReadData** führt eine Datenbankabfrage die im Attribut *Query* hinterlegt ist aus. Das Ergebnis der Abfrage wird in der Platzhalter-Variable gespeichert, welche über das Attribut *Data* angegeben wird. Die einzelnen Ergebnis-Datensätze aus der Abfrage lassen sich anschließend mit dem ForEach-Konstrukt durchiterieren.

```
<SQLReadData Data="{@myData}" Query="" Connection="{@myConnection}" DataCount ="{@ResultCount}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

10 SQLFileDownload

Die Aktion **SQLFileDownload** speichert eine Datei aus der Datebank unter dem im Attribut *FileName* angegebenen Pfad. Mit dem Attribut *Table* wird die Tabelle angegeben und mit *Field* der Feldnamen dessen Wert ausgelesen werden soll. Über das Attribut *Where* wird der Filter auf den auszulesenden Datensatz bestimmt.

```
<SQLFileDownload FileName="" Table="" Where="" DataField="" Connection="{@myConnection}" Condition ="" Variable="{@Result}" IgnoreError="false" />
```

11 SQLFileUpload

Die Aktion **SQLFileUpload** speichert eine Datei aus einem unter dem Attribut *FileName* angegebenen Pfad in der Datenbank. Mit dem Attribut *Table* wird die Tabelle angegeben und mit *Field* der Feldnamen dessen Wert ausgelesen werden soll. Über das Attribut *Where* wird der Filter auf den auszulesenden Datensatz bestimmt. Zudem besteht die optionale Möglichkeit den Namen der Datei (Attribut: *FileNameField*) und die Dateiendung (Attribut: *ExtensionField*) in der Datenbank zu speichern.

```
<SQLFileUpload FileName="" Table="" Where="" DataField="" FileNameField="" ExtensionField="" Connection="{@myConnection}" Condition="" Variable="{@Result}" IgnoreError="false" />
```