

# XML

## Inhaltsverzeichnis

- [1 Allgemeine Attribute](#)
- [2 XMLCreateDocument](#)
- [3 XMLOpenDocument](#)
- [4 XMLSaveDocument](#)
- [5 XMLHasElement](#)
- [6 XMLGetElement](#)
- [7 XMLAddElement](#)
- [8 XMLRemoveElement](#)
- [9 XMLHasChildElements](#)
- [10 XMLHasAttribute](#)
- [11 XMLGetAttribute](#)
- [12 XMLSetAttribute](#)
- [13 XMLRemoveAttribute](#)
- [14 XMLGetDocumentContent](#)
- [15 XMLElementContent](#)
- [16 XMLSetElementContent](#)
- [17 XML](#)
- [18 JSONConvertToXML](#)

Die Aktionen aus der Kategorie XML

# 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 durchgeföhrten Aktionen enthalten. Beispiel: Condition="NOT {@ResultFileExists}"

# 2 XMLCreateDocument

```
<XMLCreateDocument Document="{@myXml}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

# 3 XMLOpenDocument

```
<XMLOpenDocument Document="{@myXml}" Content="" File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

# 4 XMLSaveDocument

```
<XMLSaveDocument Document="{@myXml}" File="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

# 5 XMLHasElement

```
<XMLHasElement Document="{@myXml}" Path="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 6 XMLGetElement

```
<XMLGetElement Document="{@myXml}" Data="" Path="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 7 XMLAddElement

```
<XMLAddElement Document="{@myXml}" ParentData="" Data="" Name="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 8 XMLRemoveElement

```
<XMLRemoveElement Document="{@myXml}" Data="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 9 XMLHasChildElements

```
<XMLHasChildElements Document="{@myXml}" Path="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 10 XMLHasAttribute

```
<XMLHasAttribute Document="{@myXml}" Data="" Path="" Attribute="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 11 XMLGetAttribute

```
<XMLGetAttribute Document="{@myXml}" Data="" Path="" Attribute="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 12 XMLSetAttribute

```
<XMLSetAttribute Document="{@myXml}" Data="" Path="" Attribute="" Value="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 13 XMLRemoveAttribute

```
<XMLRemoveAttribute Document="{@myXml}" Data="" Path="" Attribute="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 14 XMLGetDocumentContent

```
<XMLGetDocumentContent Document="{@myXml}" Condition="" Variable="{@Result}" IgnoreError="false" />
```

## 15 XMLElementContent

```
<XMLGetElementContent Document="{@myXml}" Data="" Condition="" Variable>{@Result} IgnoreError="false" />
```

## 16 XMLSetElementContent

```
<XMLSetElementContent Document="{@myXml}" Data="" Content="" Condition="" Variable="@Result" IgnoreError="false" />
```

## 17 XML

Der Tag **XML** bietet die Möglichkeit direkt in dem Batchpad-Skript ein XML-Dokument zu hinterlegen. Das Dokument kann anschließend in der Form {@XML:Test} angesprochen werden. Wobei in dem Beispiel die Zeichenfolge "Test" dem Attribut *Name* entsprechen muss.

```
<XML Name="">  
  
</XML>
```

## 18 JSONConvertToXML

```
<JSONConvertToXML Document="{@myXml}" Json="" Condition="" Variable="@Result" IgnoreError="false" />
```