

Kryptographie

Inhaltsverzeichnis

- [1 Allgemeine Attribute](#)
- [2 GetMD5Hash](#)
- [3 GetSHA1Hash](#)
- [4 EncryptAES](#)
- [5 DecryptAES](#)
- [6 EncodeBase64](#)
- [7 DecodeBase64](#)
- [8 GetRandomKey](#)

Die Aktionen aus der Kategorie Kryptographie

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 GetMD5Hash

Die Aktion **GetMD5Hash** gibt einen MD5 Hashcode von einem Wert (Attribut: *Value*) zurück (Attribut: *Variable*).

```
<GetMD5Hash Value="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

3 GetSHA1Hash

Die Aktion **GetSHA1Hash** gibt einen SHA1 Hashcode von einem Wert (Attribut: *Value*) zurück (Attribut: *Variable*).

```
<GetSHA1Hash Value="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

4 EncryptAES

Die Aktion **EncryptAES** gibt einen mit AES verschlüsselten Wert von einem Wert (Attribut: *Value*) zurück (Attribut: *Variable*). Zur Verschlüsselung wird der in dem Attribut *Key* hinterlegte Schlüssel verwendet.

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

5 DecryptAES

Die Aktion **DecryptAES** gibt einen entschlüsselten Wert von einem mit AES verschlüsselten Wert (Attribut: *Value*) zurück (Attribut: *Variable*). Zur Entschlüsselung wird der in dem Attribut *Key* hinterlegte Schlüssel verwendet.

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

6 EncodeBase64

Die Aktion **EncodeBase64** gibt einen mit Base64 kodierten Wert von einem Wert (Attribut: *Value*) zurück (Attribut: *Variable*).

```
<EncodeBase64 Value="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

7 DecodeBase64

Die Aktion **DecodeBase64** gibt einen dekodierten Wert von einem mit Base64 kodierten Wert (Attribut: *Value*) zurück (Attribut: *Variable*).

```
<DecodeBase64 Value="" Condition="" Variable="{@Result}" IgnoreError="false" />
```

8 GetRandomKey

Die Aktion **GetRandomKey** gibt einen zufälligen Schlüssel zurück, der aus den Ziffern 0-9 und den Zeichen A-F besteht. Die Länge des Schlüssels wird über das Attribut *Length* bestimmt.

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