

---

**EasyScript**  
**Dokumentation für Konfiguratoren**

**appeleon**

Stand: 12. Mai 2015, Version 1.1.2

# 1 Inhaltsverzeichnis

1	Inhaltsverzeichnis .....	2
2	Allgemein .....	3
3	Berechnungen .....	4
4	Zeichenketten .....	5
5	Tabellen.....	6
6	Relationen.....	7
7	Erzeugen von aktuellem Datum/Zeitstempel .....	8
8	Benutzerinformationen.....	9
9	Objektinformationen .....	10
10	Zugriff auf Anzeigewerte .....	11

## 2 Allgemein

Es ist möglich, den Wert eines Feldes durch einfache Skripte zu berechnen.

Hierfür sind keine Programmierkenntnisse erforderlich, Erfahrungen mit Formeln in Tabellenkalkulationsprogrammen sind hilfreich.

Für die Berechnungen können Inhalte anderer Felder oder frei wählbare Texte/Zahlen verwendet werden. Zum Beispiel kann man den Wert eines Feldes zu dem Wert eines anderen Feldes addieren oder die Zahl in einem Feld mit einem Text wie etwa „Ergebnis“ ergänzen.

Um in einem Skript auf ein Feld zuzugreifen wird der Code des Feldes (ATT-Code) benötigt. Um diesen zu erhalten, wählen Sie das gewünschte Feld aus. Im Konfigurationsbaum unter "Ausgewählt" wird hinter dem Namen des Feldes der Code angezeigt.

Im Folgenden sind Zeichenfolgen in spitzen Klammern (<>) als Platzhalter zu verstehen. Beispielsweise können Sie statt **<WERT>** eine konkrete Zahl oder einen Feldtyp-Code verwenden.

Kommazahlen müssen in der englischen Schreibweise eingegeben werden, d.h. mit einem Punkt statt einem Komma: **123 . 45**

### 3 Berechnungen

Neben den Grundrechenarten Addieren, Subtrahieren, Multiplizieren und Dividieren (+ - \* /), steht auch die Berechnung von Potenzen (\*\*) und Modulo (%) zur Verfügung.

Sie können Ihre Ergebnisse durch die Funktion **round(<WERT>)** runden.

Weitere mathematische Funktionen:

Die Quadratwurzel eines Wertes ermitteln Sie durch die Funktion: **sqrt(<WERT>)**

Um einen Absolutbetrag zu erhalten, verwenden Sie die Funktion: **abs(<WERT>)**

Minimum- und Maximumwerte berechnen Sie mit den folgenden Funktionen: **min(<WERT1>, <WERT2>);**  
**max(<WERT1>, <WERT2>)**

Um mit dem Logarithmus zu rechnen, verwenden Sie die Funktion: **log(<WERT>)**, den Logarithmus zur Basis 10 berechnen Sie mit: **log10(<WERT>)**

Um eine Zufallszahl zwischen 0 und 1 zu erzeugen, verwenden Sie die **random()** Funktion.

## 4 Zeichenketten

Die Inhalte mehrerer Textfelder können verkettet werden. Hierzu werden die einzelnen Codes der Felder mit einem + verbunden.

Neben den Inhalten von vorhandenen Texten können zusätzliche Texte in Anführungszeichen angehängt werden. Sollen die verketteten Inhalte in mehreren Zeilen stehen, kann durch " \n" ein Zeilenumbruch ausgelöst werden.

Zeichenketten kürzen:

Mit der Funktion `substr(<WERT>, <Anzahl Zeichen>)` und `substr(<WERT>, <Start>, <Ende>)` können Zeichenketten verkürzt werden.

Beispiele:

```
substr("hallo", 1) - "h"
```

```
substr("hallo", 3) - "hal"
```

```
substr("hallo", 1, 3) - "hal"
```

```
substr("hallo", 2, 3) - "al"
```

## 5 Tabellen

Die Zahl der Einträge in einem Tabellenfeld kann mit `count(<Tabellen-Feld-Code>)` abgefragt werden. Dieses Ergebnis kann in arithmetischen Operationen oder in Zeichenketten verwendet werden.

Es können auch Inhalte einer Tabelle abgefragt werden: `<Tabellen-Feld-Code>[<WERT>].<Feld-Code>`. `<WERT>` gibt hierbei die Zeile an, über den `<Feld-Code>` des Feldes wird die Spalte definiert.

Beispiel: `ATT_ORDER_TABLE[1].ATT_PRICE` liefert den Preis des Objekts in der ersten Zeile (bei entsprechender Konfiguration der Felder).

### Minimum/Maximum/Summe

In Tabellen kann mithilfe der Methoden `min(<Tabellen-Feld-Code>.<Feld-Code>)`, `max(<Tabellen-Feld-Code>.<Feld-Code>)` das Minimum und Maximum gesucht werden.

Die Funktion `sum(<Tabellen-Feld-Code>.<Feld-Code>)` liefert die Summe aller Werte in einer Spalte (leere Felder werden hierbei als 0 gewertet). Diese Funktionen können mit Ganzzahlen und Kommazahlen genutzt werden.

## 6 Relationen

Objekte, die über Beziehungen mit dem aktuellen Objekt verbunden sind, können mit der Funktion `findRelatedObjects()` referenziert werden.

Beispiel: `findRelatedObjects() [1].ATT_NAME` liefert den Namen des ersten verbundenen Objekts.

Die Suche kann durch Parameter eingeschränkt werden: `<Relation-Kategorie>`, `<Objekttyp-Code>`, `<Richtung der Beziehung>`

Mit der Beziehungsrichtung gibt man an, welche Rolle das aktuelle Objekt (an dem EasyScript ausgeführt wird) in dieser Beziehung hat.

Parameter, die nicht angegeben werden, werden nicht beachtet. Gibt man also `SOURCE` bzw. `TARGET` nicht an, wird eine Liste von Objekten zurückgeliefert, bei denen das aktuelle Objekt sowohl `SOURCE` als auch `TARGET` der Beziehung sein kann.

Beispiele:

`findRelatedObjects(REC_IS_CHILD, SOURCE)` - Liefert alle Objekte, die über die Kategorie `REC_IS_CHILD` verbunden sind und das aktuelle Objekt `SOURCE` (Quelle) der Beziehung ist.

`findRelatedObjects(OBT_CONTACT)` – Liefert alle Kontakte, die mit dem aktuellen Objekt eine Beziehung haben.

`findRelatedObjects(OBT_CONTACT, REC_IS_CHILD, SOURCE)` - Kombination aus den obigen Beispielen.

## 7 Erzeugen von aktuellem Datum/Zeitstempel

Die Funktion `now()` liefert das aktuelle Datum und die aktuelle Uhrzeit zurück. Mit der Funktion `format(<Formatierung>, (Optional <Sprache>))` kann das Datum entsprechend formatiert werden. (Geeignet für die Verwendung in Text- oder Langtextfeldern, bei denen die normale Formatierungsfunktion nicht verfügbar ist.)

Parameter:

**<Formatierung>**

Zeichen	Beschreibung	Beispiel
G	Ära-Kennung	AD
y	Jahr	1996 ; 96
M	Monat im Jahr	July ; Jul ; 07
w	Woche im Jahr	27
W	Woche im Monat	2
D	Tag im Jahr	189
d	Tag im Monat	10
F	Tag der Woche im Monat	2
E	Tag in der Woche	Tuesday ; Tue
a	AM / PM-Marker	PM
H	Stunden im Tag (0-23)	0
k	Stunden im Tag (1-24)	24
K	Stunde in am/pm (0-11)	0
h	Stunde in am/pm (1-12)	12
m	Minute der Stunde	30
s	Sekunde in der Stunde	55
S	Millisekunde	978
z	Allgemeine Zeitzone	Pacific Standard Time ; GMT-08:00
Z	RFC 822 Zeitzone	-0800

Formatierung von Text (z.B. Wochentag): Wird der Musterbuchstabe vier Mal hintereinander verwendet, so wird die Langform verwendet (z.B. "Dienstag") andernfalls die Kurzform ("Die"), falls verfügbar.

Formatierung von Zahlen: Die Anzahl der Musterbuchstaben bestimmt die Stellen der Zahl. Hat die Zahl, die ausgegeben werden soll, weniger Stellen, werden sie mit Nullen aufgefüllt.

Formatierung von Jahreszahlen: Bei der Verwendung von zwei Musterbuchstaben wird das Jahr auf zwei Stellen gekürzt (z.B. 2004 -> 04). Ansonsten ist das Verhalten genauso wie bei der Formatierung von Zahlen.

**<Sprache>** - Zweistelliger Sprachcode, mit dem das Datum formatiert werden soll (z.B. "en" oder "de")

Beispiele:

`now().format("EEEE")` - Liefert den Wochentag als Wort (z.B. "Dienstag")

`"Es ist jetzt " + now().format("HH:mm:ss") + " Uhr."` - Ergibt "Es ist jetzt 17:05:12 Uhr".

`now().format("EEEE", "en")` - "Tuesday"

`now().format("EEEE", "de")` - "Dienstag"



## 8 Benutzerinformationen

Es ist möglich, verschiedene Informationen über einen Benutzer abzufragen.

Folgende Variablen mit Benutzerinformationen sind vorhanden:

**user** - aktuell angemeldeter Benutzer, der das Objekt geöffnet hat

**create\_user** - Benutzer, der das Objekt erstellt hat

**update\_user** - Benutzer, der das Objekt zuletzt verändert und gespeichert hat

Folgende Informationen können abgefragt werden:

**id** - Id des Benutzers

**language** - Sprache, in der der Benutzer gerade angemeldet ist (nur für **user** verfügbar)

**username** - Benutzername

**userDetail** - das Objekt, welches mit dem Benutzer verknüpft ist

**name** - Name des Benutzers, wie er in seinem Profil festgelegt ist

**firstname** - Vorname des Benutzers

**email** - E-Mail Adresse des Benutzers

Beispiele:

`user.language` - "de"

`createUser.userDetail.ATT_NAME` - "Müller"

`updateUser.name` - "Mayer"

## 9 Objektinformationen

Man kann über folgende Variablen auf Objektinformationen zugreifen:

`id` - ID des Objekts

`createUser` - Benutzer, der das Objekt erstellt hat (s. Benutzerinformationen)

`createDate` - Datum, an dem das Objekt erstellt wurde

`updateUser` - Benutzer, der das Objekt zuletzt verändert und gespeichert hat (s. Benutzerinformationen)

`updateDate` - Datum der letzten Änderung

`openedByObject` - Objekt, von dem aus das aktuelle Objekt geöffnet wurde

(nicht verfügbar, wenn das Objekt im Anzeigemodus geöffnet wird)

Beispiel: Man erstellt ein neues Objekt durch Rechtsklick auf eine Zeile in einer Tabelle.

Das `openedByObject` ist dann das Objekt, auf dem der Rechtsklick ausgeführt wurde.

Beispiele:

`createUser.name` - "Müller"

`updateDate.format("EEEE")` - "Dienstag"

`openedByObject.ATT_NAME` - "Testobjekt"

`openedByObject.createUser.userDetail.ATT_NAME` - "Mayer"

## 10 Zugriff auf Anzeigewerte

Ist ein Feld als Auswahlfeld konfiguriert, kann über das Attribut **key** auf den Wert zugegriffen werden, der im Feld gespeichert ist.

Mit **value** bzw. **displayValue** kann der Anzeigewert ermittelt werden.

Der Anzeigewert kann beispielsweise auch bei Datumsfeldern mit Formatierung abgefragt werden.

Beispiele:

```
ATT_TYPE.key - "PRIO0010"
```

```
ATT_TYPE.value - "B"
```

```
ATT_DATE.displayValue - "01.02.2011"
```