Funktionen im Feldeditor (Regeln)

Im Feldeditor können - wie im Druckdesigner - Operatoren und Funktionen genutzt werden.

Beachten Sie:

Diese Funktionen werden je nach Verwendungszweck (Exportfilter, Bedingungen in Regeln) mit einem **$-Zeichen** an erster Stelle eingegeben.

Sofern eine Funktion verwendet wird, die **parametrisiert** werden kann, muss dies in der **Regelbedingung** anders als in einer **Regelanweisung** eingegeben werden.

Beispielsweise kann die Funktion "**GetAktDate()**" z. B. folgende **Parameter** enthalten: "**Time**" oder auch "**Server, Time**".

Regel-Bedingung	Regel-Anweisung
GetAktDate(**'Time'**)	**$**GetAktDate(**Time**)
GetAktDate(**'Server', 'Time'**)	**$**GetAktDate(**Server, Time**)

Inhalt

Folgende Operatoren können verwendet werden:

1. Vergleichs-Operatoren:

- **and**:
  - **(a) and (b)**
    - a und b müssen Wahr sein
    - («SollBetLW1» = «HabenBetLW1») and («SollBetLW2» = «HabenBetLW2»)
- **or**:
  - **(a) or (b)**
    - a oder b muss Wahr sein
    - («SollBetLW1» = «HabenBetLW1») or («SollBetLW2» = «HabenBetLW2»)
- **<>:**
  - **(a) <> (b)**
    - Wenn a sich von b unterscheidet ist das Ergebnis Wahr
    - «AdrNr» <> '10000'

2. Verneinungs-Operator (Negation):

- **not:**
  - **not (a)**
    - a muss FALSCH sein, um als Ergebnis Wahr zu erhalten
  - **not («SollBetLW1» = «HabenBetLW1»)**
    - SollBetLW1 und HabenBetLW1 müssen sich unterscheiden, um als Ergebnis Wahr zu erhalten

3. Mengen-Operator:

- **in:**
  - **(a in (b))**
  - **(Wert in (Werte-Menge))**
    - Wenn a (Wert) in b (Werte-Menge) vorliegt ist das Ergebnis Wahr
  - **«AdrNr» in ('10000', '10001')**
    - Dies ist eine absolute Definition.
    - Es werden nur durch Komma Werte gefunden (durch Komma separierte Werte müssen nach jedem Komma ein Leerzeichen besitzen, damit diese ausgewertet werden können).
    - Ein Bereich kann nicht mit dem Operator "in" abgefragt werden.

Tipp:

Wollen Sie einen "**von - bis**" Bereich, so können Sie folgende Bedingung definieren:

Pos('100',«AdrNr»)>0

Ein mit dieser Formel definierter Filter zeigt alle Adressen an, die '100' enthalten. (Vgl. Sie auch mit der Beschreibung der Funktion **Pos()** weiter unten).

Folgende Funktionen können in den Filter- bzw. in den Regelbedingungen genutzt werden:

Text / String-Funktionen:

- **ToString([Wert]) oder bei Filter auch mal ToText([Wert]) möglich**
  - Wandelt den Wert in einen String um
- **Left([String],[Anzahl der Zeichen])**
  - **Left(a,n)**
    - Die ersten n Zeichen von dem String a werden zurückgegeben
  - **Left('TEST',2)**
    - Die ersten 2 Zeichen von dem String: "TEST" werden zurückgegeben. Ergebnis: "TE"
  - **Left(«AdrNr»,3) = '100'**
    - Die ersten 3 Zeichen von der String-Variable: "AdrNr (Adressnummer)" werden mit dem Wert: "100" verglichen. Das Ergebnis ist positiv, sofern die AdrNr (Adressnummer) mit einer 100 beginnt.
- **Right([String],[Anzahl der Zeichen])**
  - **Right(a,n)**
    - n Zeichen von rechts von dem String a werden zurückgegeben
  - **Right('TEST',2)**
    - Die letzten 2 Zeichen von dem String: "TEST" werden zurückgegeben. Ergebnis: "ST"
  - **Right(«AdrNr»,3) = '100'**
    - Die letzten 3 Zeichen von der String-Variable: "AdrNr (Adressnummer)" werden mit dem Wert: "100" verglichen. Das Ergebnis ist positiv, sofern die AdrNr (Adressnummer) mit einer 100 endet.
- **Mid([String],[Start-Position],[Anzahl der Zeichen])**
  - **Mid(a,n1,n2)**
    - Ab der Position n1 werden n Zeichen von dem String a zurückgegeben
  - **Mid("TEST",2,2)**
    - Ab der Position 2 werden 2 Zeichen von dem String "TEST" zurückgegeben. Ergebnis: "ES"
  - **Mid(«AdrNr», 3, 3) = '100'**
    - Ab der Position 3 werden 3 Zeichen von der String-Variable: "AdrNr (Adressnummer)" mit dem Wert: "**100**" verglichen
    - Beachten Sie, dass nach dem Komma jeweils ein Leerzeichen angegeben wird, ansonsten kann die Formel nicht ausgewertet werden
    - Beispiel mit oben genannter Formel: Das Ergebnis ist dann positiv, sofern die AdrNr (Adressnummer) z. B. "10**100**" ist, da ab Position 3 die Bedingung für 3 Zeichen wahr ist.
- **Pos([Such-String],[String])**
  - Gibt die Position von dem gesuchten String innerhalb eines Strings zurück.
  - **Pos("c","abc")**
    - Sucht "c" innerhalb von "abc" und gibt die Position zurück. Das Ergebnis ist: "3"

Beachten Sie:

Eine Pos-Funktion in einem E-Mail-Layout gibt immer den Rückgabewert "Leer" zurück, sofern der gesuchte Text nicht enthalten ist.

Sofern man eine Pos-Funktion in einer Cond - / If Bedingung verwendet, muss man einen Vorgabewert / Vergleichswert mit übergeben (als 3. Angabe). Dieser Vorgabewert wird dann zurückgegeben, sofern der gesuchte Text nicht verfügbar ist.

Beispiele:

- Allgemein:
  - $Cond($Pos([Suchwort],[Feld],0)<>0,JA,NEIN)
- Vorgangsart: Angebot - Layout: Angebots-Mail - Feld: Empfänger-Formel:
  - $Cond($Pos("AN",«BelegNr»,0)<>0, true@microtech.de, false@microtech.de)
    - Liefert die E-Mailadresse: "true@"
  - $Cond($Pos("AG",«BelegNr»,0)<>0, true@microtech.de, false@microtech.de)
    - Liefert die E-Mailadresse: "false@"
  - $Cond($Pos("AG",«BelegNr»,0)<>0, true@microtech.de)
    - Liefert keine E-Mailadresse bzw. würde dann als Empfänger-Mailadresse einfach nichts ("Leer") eintragen.

- **Length([String])**: XL
    - Die Zeichenlänge eines Strings werden zurückgeliefert
    - **Length("Ich bin ein Test!")**
        - Das Ergebnis liefert den Wert "17" zurück.
- **GetAktBzr():**
    - Das Benutzerkürzel des aktuellen Benutzers wird zurückgeliefert

Date / Time-Funktionen:

- **Date([Wert]) oder ToDate([Wert])**
    - Wandelt den Wert in ein Datum um
    - **Date('01.01.80')**
        - Der Wert "01.01.80" kann mit einem Datumsfeld verglichen werden
- **Time([Wert]) oder ToTime([Wert])**
    - Wandelt den Wert in eine Uhrzeit um
    - **Time('08:30:00')**
        - Der Wert "08:30:00" kann mit einen Zeitfeld verglichen werden
- **DateTime([Wert]) oder ToDateTime([Wert])**
    - Wandelt den Wert in ein Datum mit Uhrzeit um
    - **DateTime('01.01.80 08:30:00')**
        - Der Wert "01.01.80 08:30:00" kann mit einen DatumZeitfeld verglichen werden
- **Year([Date oder DateTime])**
    - Die Jahreszahl wird zurückgeliefert
- **Month([Date oder DateTime])**
    - Der Monat wird zurückgeliefert
- **Day([Date oder DateTime])**
    - Der Tag innerhalb des Monats wird zurückgeliefert
- **Hour([Date oder DateTime])**
    - Die Stunde wird zurückgeliefert
- **Minute([Date oder DateTime])**
    - Die Minute wird zurückgeliefert
- **Second([Date oder DateTime])**
    - Die Sekunde wird zurückgeliefert
- **GetDate()**
    - Das aktuelle Systemdatum wird zurückgeliefert
- **GetAktDate()**
    - Das aktuelle Programmdatum wird zurückgeliefert
    - **GetAktDate('Time'):** L
        - Das aktuelle Arbeitsdatum mit Uhrzeit wird zurückgeliefert.
    - **GetAktDate('Server'):** L
        - Das aktuelle Serverdatum (Datum des Server-PCs) wird zurückgeliefert.
    - **GetAktDate('Server, Time'):** L
        - Das aktuelle Serverdatum mit Uhrzeit (Datum des Server-PCs) wird zurückgeliefert.
- **DiffDate([Start-Datum],[End-Datum])** XL
    - Ermittelt die Anzahl der Tage zwischen zwei Daten
    - Als Rückgabewert erhält man eine ganzzahlige Anzahl der Tage.
        - **DiffDate(1.1.2022,2.1.2022)**
            - Das Ergebnis ist: "**1**"
        - **DiffDate(1.1.2022,1.1.2022)**
            - Das Ergebnis ist: "**0**"
        - **DiffDate(2.1.2022,1.1.2022)**
            - Das Ergebnis ist: "**-1**"
    - **Beispiele:**
        - **DiffDate(«VPo.LiefDat»,GetAktdate()) > 3**
            - Es soll bei der Erfassung einer Vorgangsposition geprüft werden, ob das Lieferdatum der Position größer als das aktuelle Datum + 3 ist.
        - **DiffDate(«LiefDat»,Getaktdate()) > 2**
            - In einer Vorgangsliste sollen nur Vorgänge ausgegeben werden deren Lieferdatum älter als 2 Tage als das aktuelle Datum sind.

Boolean (Kennzeichen)-Funktion:

- **ToBool([Wert])**
  - Wandelt den Wert in ein Boolean (Ja/Nein) um

Integer / Float (Zahlen / Fließkommazahl)-Funktionen:

- **ToInt([Wert])**
  - Wandelt den Wert in eine Ganzzahl um
- **ToFloat([Wert])**
  - Wandelt den Wert in eine Fließkommazahl um

Sonder-Funktion:

- **CalcUmsatz([Anzahl der letzten Monate])**
  - Berechnet den Umsatz der letzten angegebenen Monate

DBInfo-Funktion:

- **DBInfo('Name der Datentabelle', Schlüsselwert[e],'Zielfeldname')**
  - Findet in der Datentabelle "**Name der Datentabelle**" den Datensatz mit dem entsprechenden "**Schlüsselwert**" und gibt den Inhalt des Feldes "**Zielfeldname**" zurück.
  - Mit Hilfe der DBInfo-Formeln kann ein Wert aus einer Datentabelle unter Verwendung eines Primärindex abgefragt werden.
    - **Erläuterung zum Aufbau:**
      - **Name der Datentabelle**
        - Name der Tabelle, aus der ein Wert abgefragt werden soll.
      - **Schlüsselwert[e]**
        - Kann beliebigen Typs sein. Sollte jedoch zum 1. Feld des Primärindex der abzufragenden Tabelle passen.
      - **Zielfeldname**
        - Optionaler Name des Feldes, dessen Wert zurück geliefert werden soll. Falls dieser Parameter ausgelassen wird, wird dieselbe Zeichenfolge zurückgegeben mit der der Datensatz z. B. beim Löschen bezeichnet wird.
  - Die DBInfo-Formel steht Ihnen auch im Feldeditor zur Verfügung. Der inhaltliche Aufbau ist wie im DBInfo-Formeln für Bereichsfilter und Ausgabefilter.
  - **Beispiel:**
    - **DBInfo('Adressen',«AktVog.AdrNr»,'SteuNr')**
      - Dadurch wird z.B. in einer Vorgangsliste das Feld Steuernummer der Adresse abgefragt.

Weitere Themen

- Beispiele für Funktionen im Feldeditor (Regeln)