

# Funktionen im Kassensbondruck

Innerhalb der Kassensbondrucke können Funktionen eingebunden werden, um die Druckausgabe noch individueller zu gestalten.

Mit den Funktionen \$COND bzw. \$IF können Sie sich auf einfache Weise selbst eine WENN - DANN - SONST Bedingungen formulieren:

WENN <erstes Argument> DANN <zweites Argument> SONST <drittes Argument>.

\$Cond (Bedingung, Rückgabe wenn Bedingung wahr [,Rückgabe wenn Bedingung falsch])

\$If (Bedingung, Rückgabe wenn Bedingung wahr [,Rückgabe wenn Bedingung falsch])

Die ersten beiden Parameter sind Pflicht. Der dritte ist optional. Parameter die ein Komma, eine schließende Klammer oder ein Anführungszeichen enthalten, müssen in Anführungszeichen eingeschlossen werden. Eventuell enthaltene Anführungszeichen müssen verdoppelt werden.

Als Bedingung sind einfache mathematische Ausdrücke erlaubt:

= , <>, >, >=, <, <=, Klammersetzung, and, or, not, is, null,

Ein Text muss zum Vergleichen mit einem Feld in einfache Anführungszeichen eingeschlossen werden ( '123' ).

Um beispielsweise zu überprüfen, ob das Feld KPo.Bez den Inhalt >> Test ("2,x") Test << enthält, kann folgender Ausdruck verwendet werden:

\$Cond("«KPo.Bez[;25]»='Test ('"2,x"') Test' ",Ja,Nein)

Nachfolgend ist ein Beispiel aufgeführt, um im Ausdruck des Kassensbelegs nur dann den Rabatt auszuweisen, wenn die entsprechende Position auch Rabatt enthält:

«KPo.PosNr[R;3]» «KPo.ArtNr[;25]»\r\n

«KPo.Bez[;25]»\r\n

«KPo.Mge[R;5]»\*«KPo.EPreis[R;9]» «Kas.Waehr[;3]» = «KPo.Preis[R;9]» «Kas.Waehr[;3]»\r\n

\$Cond("(«KPo.RabSz»<>0) and not («KPo.RabSz» is null)", «KPo.RabSz» % Rabatt = «KPo.RabBet» «Kas.Waehr[;3]»\r\n)

Folgende zusätzliche Funktionen können in diesem Bereich genutzt werden:

|                              |   |
|------------------------------|---|
| ToString(Wert)               | wandelt in Text um                      |
| ToInt(Wert)                  | wandelt in Ganzzahl um                  |
| ToBool(Wert)                 | wandelt in Boolean (Ja/Nein) um         |
| ToFloat(Wert)                | wandelt in Fließkommazahl um            |
| ToDate(Wert) oder Date(Wert) | wandelt in Datum um                     |
| Time(Wert) oder Time(Wert)   | wandelt in Zeitangabe um                |
| DateTime(Wert)               | wandelt in Datums-/ Zeitangabe um       |
| Year(Date oder DateTime)     | gibt die Jahreszahl zurück              |
| Month(Date oder DateTime)    | gibt den Monat zurück                   |
| Day(Date oder DateTime)      | gibt den Tag innerhalb des Monat zurück |
| Hour(Time oder DateTime)     | gibt die Stunde zurück                  |
| Minute(Time oder DateTime)   | gibt die Minuten zurück                 |
| Second(Time oder DateTime)   | gibt die Sekunden zurück                |
| GetDate()                    | gibt das aktuelle Datum zurück          |

## Beispiel: Ausgabe des Vor- und Nachnamen des aktuellen Verkäufers auf einem Kassensbon

Mit Hilfe der nachfolgenden Formeln können Sie auf die Benutzer-Daten des jeweiligen Verkäufers zugreifen. Die erste Formel weist den Vornamen, die zweite den Nachnamen aus:

\$DBInfo(Benutzer,IX;KennZ;1,«Kas.VerK»,VNa)

\$DBInfo(Benutzer,IX;KennZ;1,«Kas.VerK»,NNa)

Sind keine Benutzerdaten hinterlegt, kann auf Grund der vergebenen Verkäufer-Nummer eine \$Cond-Abfrage einen festen Wert ausgeben:

\$Cond(«Kas.Verk» = '1', Frau Test, \$Cond(«Kas.Verk» = '2', Herr Probe, \$Cond(«Kas.Verk» = '3', Frau Aushilfe)