

# Wie vergleicht man Werte im Makrocode richtig?

## Beschreibung:

In Makros kommt es ständig vor, dass man einen Wert mit einem anderen vergleicht z.B. in einem Skript wird dem Benutzer mehrere Optionen angeboten was der Gesprächspartner geantwortet hat. Diese werden dann letztendlich im Makro ausgewertet (verglichen) um anschließend die gewünschte Aktion durchzuführen.

## Problem:

Für uns ist ein Vergleich von z.B. ist  $1 = 1$  eindeutig klar und zwar, dass die beiden Werte identisch sind.

Doch für den Compiler, welcher das Makro verarbeitet, ist  $1 = 1$  nicht unbedingt identisch.

## Ursache:

Der Compiler berücksichtigt nicht nur den Wert alleine, sondern auch den Datentyp.

## Automatische Konvertierung:

Der Compiler versucht selbstständig einen Vergleich von zwei Werten mit unterschiedlichen Datentypen auf einen gleichen Datentyp zu konvertieren.

Dies gelingt ihm in soweit, wenn die zwei Werte in einen gleichen Datentyp konvertiert werden können und die Werte statisch sind.

Statisch bedeutet, dass der Wert in keinem Feld oder einer Variable gespeichert ist sondern fix im Makro definiert ist.

## Beispiel:

Wir speichern den Wert 1 als Ganzzahl in der Variable vInt und als Text in der Variable vText (Anführungszeichen leiten einen Text ein):

vInt=1 (Datentyp Integer)

vText="1" (Datentyp String)

Nun machen wir unterschiedliche Vergleiche auf die Variable vInt, welche eine Zahl vom Datentyp Integer enthält:

## Variante 1:

```
If vInt=1 Then
MsgBox "Wahr"
Else
MsgBox "Falsch"
End If
```

Ergebnis = Wahr (Keine Konvertierung erforderlich da beide Werte den Datentyp Integer haben)

## Makro-Code

### Variante 2:

```
If vInt="1" Then  
MsgBox "Wahr"  
Else  
MsgBox "Falsch"  
End If
```

Ergebnis = Wahr (eine automatische Konvertierung hat statt gefunden da der Datentyp Integer ungleich String ist)

### Variante 3:

```
If vInt=vText Then  
MsgBox "Wahr"  
Else  
MsgBox "Falsch"  
End If
```

Ergebnis = Falsch (Die automatische Konvertierung greift nicht, da beide Werte in einer Variable gespeichert sind)

### Variante 4:

Verwendung eines auf der Adresstabelle angelegten Feldes "Summe" vom Datentyp Ganzzahl:

```
If ActiveRecord.Fields("Summe").Value = 1 Then  
MsgBox "Wahr"  
Else  
MsgBox "Falsch"  
End If
```

Ergebnis = Wahr

Da der Datentyp Ganzzahl in AG-VIP SQL ein Long Datentyp darstellt, wird eine automatische Konvertierung durchgeführt da die Zahl 1 vom Datentyp Integer ist.

### Variante 5:

Der Vergleich von zwei Feldern welche in AG-VIP SQL angemeldet sind oder ein Vergleich zwischen einer Variable und einem AG-VIP SQL Feld ergeben das gleiche

Ergebnis wie bei der Variante 3 **Makro-Center** -> keine Umwandlung.

### **Richtige Vorgehensweise:**

Variablen und Felder sollten immer manuell konvertiert werden.  
Dafür gibt es in VBScript Konvertierungsfunktionen.

Hier eine Liste der gebräuchlichsten:

- CBool() -> Umwandlung in ein logischen Datentyp (Boolean)
- CDate() -> Umwandlung in ein Datums Datentyp (Date)
- CDbl() -> Umwandlung in ein Fließkomma Datentyp (Double)
- CInt() -> Umwandlung in ein Ganzzahl Datentyp 16bit (Integer)
- CLng() -> Umwandlung in ein Ganzzahl Datentyp 32bit (Long)
- CStr() -> Umwandlung in ein Text Datentyp (String)

Mit Hilfe dieser Funktionen kann die Variante 3 ebenfalls wie folgt verwendet werden:

```
If CStr(vInt)=CStr(vText) Then
MsgBox "Wahr"
Else
MsgBox "Falsch"
End If
```

Ergebnis = Wahr

Oder wenn sichergestellt ist, dass in der Variable vText eine Zahl gespeichert ist auch folgende Konvertierung:

```
If CLng(vInt)=CLng(vText) Then
MsgBox "Wahr"
Else
MsgBox "Falsch"
End If
```

Ergebnis = Wahr

### **Hinweis bei Vergleichen von Texten (Strings):**

Bei einem Vergleich von Text Feldern bzw. Variablen sollte man immer sofern man dies nicht berücksichtigen möchte die Leerzeichen am Anfang und am Ende des Wertes entfernen.

Dafür gibt es die VBScript Funktion Trim() welche im folgenden Beispiel verwendet wird:

## Makro-Center

```
If Trim(CStr(vInt))=Trim(CStr(vText)) Then  
MsgBox "Wahr"  
Else  
MsgBox "Falsch"  
End If
```

Ergebnis = Wahr

### Wichtiger Hinweis:

Die Konvertierungsfunktion CInt() sollte wenn möglich nicht verwendet werden. In AG-VIP SQL gibt es den Ganzzahl Datentyp Integer für angelegte Felder nicht sondern nur den Ganzzahl Datentyp Long.

Würde man nun versuchen eine Variable vom Datentyp Long in einen Integer umzuwandeln, dann würde das solange funktionieren bis der Wert der Variable den 16bit Wertebereich des Integers überschreitet (-32768 bis 32767). Eine Zahl die also kleiner -32768 oder größer 32767 ist, könnte nicht in einen Integer konvertiert werden und würde deshalb einen Makrofehler verursachen.

Es empfiehlt sich immer eine Konvertierungsfunktion zu verwenden, denn auch wenn man denkt, dass man weiß, dass die Datentypen in den zu vergleichenden Feldern bzw. Variablen der selbe ist, könnte eventuell irgendwann mal jemand den Datentyp ändern und Ihr Makro würde nicht mehr funktionieren.

So gab es z.B. mit der AG-VIP SQL Version 1.40.004 eine Datentyp Korrektur bei den Ids und einigen Zahlenfeldern.

Diese wurden in den vorherigen Versionen immer als Text (String) zurückgegeben und diese werden nun als Ganzzahl (Long) zurückgegeben.

### Änderungslog:

11.07.2013

- Formatierungsfehler im Artikel behoben
- Erweiterung des Artikels um den Abschnitt "Hinweis bei Vergleichen von Texten (Strings)"

Eindeutige ID: #1314

## Makro-Center

Verfasser: Peter Brandt [Grutzeck Software GmbH]

Letzte Änderung: 2013-07-11 09:52