
Einfache Dialoge

Autor & Copyright: Dipl.-Ing. Harald Nahrstedt

Version: 2016 / 2019 / 2021 / 365

Erstellungsdatum: 15.10.2011

Überarbeitung: 01.12.2023

Beschreibung:

In VBA gibt es einfache Dialogfunktionen, die einen schnellen Prozeduraufbau erlauben. Anders als die Methode *Dialogs*, sind sie mit wenigen Parametern programmierbar.

Anwendungs-Datei: AE-012_EinfacheDialoge.xlsm

1 Die Inputbox-Funktion

Nicht selten möchte man einer Variablen in einer Excel-Prozedur einen Wert zuweisen, der nicht fest vorgegeben werden soll. Eine einfache Möglichkeit ist die Inputbox. Die Syntax lautet

```
InputBox (prompt[, title] [, default] [, xpos] [, ypos] [, helpfile,  
context])
```

In eckigen Klammern stehende Parameter müssen dabei nicht gesetzt werden, sodass ein einfaches Beispiel die Form

Codeliste 1. Einfache Eingabe-Prozedur

```
Sub Test1()  
    Dim vTest As Variant  
    vTest = InputBox("Hinweistext zur Eingabe")  
End Sub
```

haben kann (Bild 1).

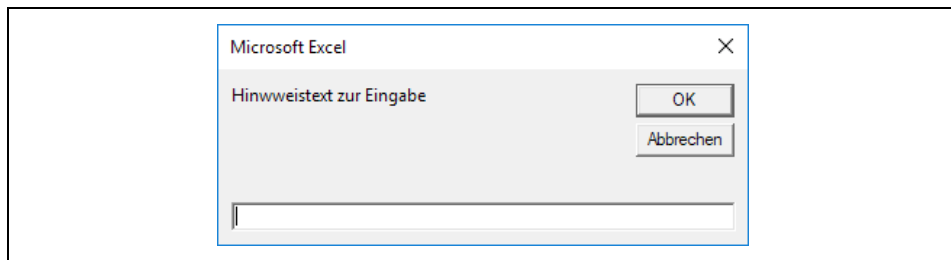


Bild 1. Input-Dialogfenster

Fällt der Hinweistext etwas länger aus, kann auch ein Zeilenvorschub eingesetzt werden. Das einfache Beispiel, bei dem auch schon ein Vorgabetext steht,

Codeliste 2. Eingabe-Prozedur mit Vorgabetext

```
Sub Test2()  
    Dim vTest As Variant  
    vTest = InputBox("Hier steht ein Text" & vbCrLf & _  
        "Hier steht noch eine Information", , "Ihr Eintrag")  
End Sub
```

sieht dann wie folgt aus (Bild 2).

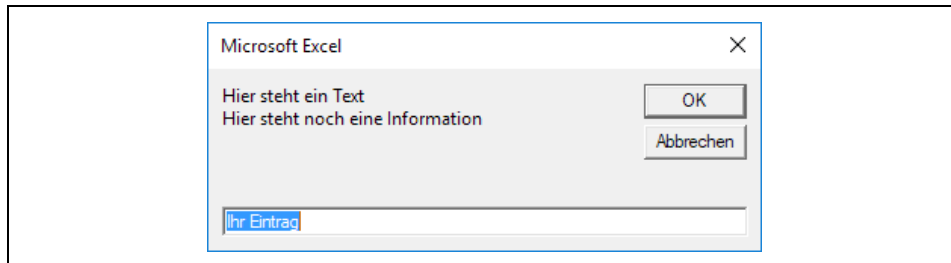


Bild 2. Input-Dialogfenster mit Vorgabetext

Über die Parameter *xpos* und *ypos* kann die Inputbox an eine gewünschte Stelle platziert werden, Die Werte *xpos=1* und *ypos=1* positionieren sie in die linke obere Ecke des Anwendungsfensters. Die Parameter *helpfile* und *context* werden eher selten gebraucht. Sie setzen eine Hilfedatei (*.chm) voraus, die mit einem entsprechenden Hilfsprogramm erstellt werden muss.

2 Die MsgBox-Methode

Die Prozedur *MsgBox* (Messagebox) dient zur Ausgabe von Informationen. Sie kann auf zwei Arten genutzt werden. Die erste Möglichkeit ist eine einfache Ausgabe von Text. Zur Bestätigung, dass die MsgBox erkannt wurde, muss ein OK-Button betätigt werden.

Die Syntax der *MsgBox* für den ersten Fall lautet

```
MsgBox(Prompt,Buttons As vbMsgBoxStyle = vbOKOnly, _  
[Titel],[HelpFile],[Context] As vbMsgBoxResult)
```

Auch hier müssen die in eckigen Klammern gesetzten Parameter nicht gesetzt werden. Prompt steht auch hier für konstanten Text oder eine Variable, deren Inhalt gezeigt werden soll. *Buttons* legt die Anzahl und Form der Buttons fest. Siehe nachfolgende Tabelle. Ebenso steht eine Auswahl von Symbolen zur Verfügung, so dass auch mehrere Buttons kombiniert werden können.

Die Verknüpfung erfolgt mit dem + Operator (nicht mit dem & Operator). Nachfolgend ein paar Beispiele.

```
MsgBox "Dieser Text wird angezeigt!"  
MsgBox Variable  
MsgBox "Dieser Text ist wichtig", vbOKOnly + vbCritical  
MsgBox "Ausgabertext", vbCritical, "Ausgabebetitel"
```

3 Die Reaktion auf eine MsgBox abfragen

Aus der Syntax der *MsgBox*-Methode wird ersichtlich, dass es sich hierbei um eine Funktion handelt. (As *vbMsgBoxResult*), die einen Wert vom Typ *vbMsgBoxResult* liefert. Dieser Wert kann abgefragt und zur weiteren Steuerung der Prozedur verwendet werden, wie im nachfolgenden Beispiel.

Codeliste 3. Eingabe-Prozedur mit Rückmeldung

```
Sub Test3()  
  Dim Result As Variant  
  Result = MsgBox("Bitte drücken Sie die OK-Taste!", _  
    vbOKCancel, "Abfrage")  
  If Result = vbOK Then  
    MsgBox "Richtig! " & _  
      "Sie haben die OK-Taste gedrückt!", vbInformation  
  Else  
    MsgBox "Falsch! " & _  
      "Das war die Abbrechen-Taste.", vbCritical  
  End If  
End Sub
```

Die nachfolgenden Tabellen zeigen noch einmal die verschiedenen Buttons und Symbole, so wie die Reaktionswerte für eine Abfrage. Außer den VB-Konstanten

können auch die dazugehörigen Binärwerte genutzt werden. So kann statt vbOKCancel + vbCritical auch der Wert: $1 + 16 = 17$ gesetzt werden.

Tabelle 1. Buttons

Wert	Konstantenname	Beschreibung
0	vbOKOnly	OK
1	vbOKCancel	OK - Abbrechen
2	vbAbortRetry	Wiederholen - Ignorieren
3	vbYesNoCancel	Ja - Nein - Abbrechen
4	vbYesNo	Ja – Nein
5	vbRetryCancel	Wiederholen - Abbrechen

Tabelle 2. Results

Wert	Konstantenname	Reaktion
1	vbOK	Klick auf OK
2	vbCancel	Klick auf Abbrechen
3	vbAbort	Klick auf Abbrechen
4	vbRetry	Klick auf Wiederholen
5	vbIgnore	Klick auf Ignorieren
6	vbYes	Klick auf Ja
7	vbNo	Klick auf Nein

Tabelle 3. Symbole

Wert	Konstantenname	Beschreibung
16	vbCritical	Kritische Meldung
32	vbQuestion	Fragezeichen
48	vbExclamation	Warnung
64	vbInformation	Information

4 Den Focus auf einen anderen Button legen

Beim Aufruf der MsgBox bekommt standardmäßig die erste Schaltfläche von links den Focus. Soll eine andere Schaltfläche den Focus bekommen, kann dies durch Addition von VB Konstanten erfolgen. Eine Möglichkeit zeigt das nachfolgende Beispiel.

Codeliste 4. Die Prozedur BeispielFocus wertet den Rückgabewert aus

```
Sub BeispielFocus()
    Dim Result As Variant
    Result = MsgBox("Bitte wählen Sie!", _
        vbYesNoCancel + vbQuestion + vbDefaultButton2, _
        "ATTENTION!!!")
```

```
If Result = vbNo Then
    MsgBox "Nein Taste gedrückt!", vbCritical
Else
    If Result = vbYes Then
        MsgBox "Ja Taste gedrückt!", vbInformation
    Else
        MsgBox "Cancel gedrückt!", vbQuestion
    End If
End If
End Sub
```

Entsprechend bekommt mit *vbDefaultButton3* der dritte Button von links den Focus.