

16.7 MaskBox

Die Klasse MaskBox (gb.form) implementiert eine TextBox mit einer Eingabe-Maske. So verlockend die Verwendung einer MaskBox auch scheint – die Masken 990.990.990.990 für eine IP-Adresse oder 09.09.0000 für ein Datum lassen auch unsinnige Eingaben wie ein Datum mit 02.13.2014 oder eine IP-Adresse mit 202.112.237.273 zu.

Die wichtigste Eigenschaft einer MaskBox ist *MaskBox.Mask*. Die Maske ist eine Zeichenkette, die vorgibt, welche Eingabe-Zeichen erlaubt sind (Eingabe-Alphabet). Die Maske kann gesetzt oder ausgelesen werden. Ist die Maske leer, dann verhält sich die MaskBox wie eine TextBox.

| Maske | | Default | Beschreibung |
|----------------------|----------------------------|-------------|--|
| [...] | | Leerzeichen | Jedes Zeichen aus dem angegebenen Bereich ist zulässig. Die Syntax für den Zeichenbereich ist der gleiche wie für den Operator LIKE → Kapitel 8.3.3. |
| ? | Jedes Zeichen | Leerzeichen | Jedes Zeichen ist erlaubt |
| 0 | Ziffer | 0 | Jede Ziffer ist erlaubt - 0 wird angezeigt. |
| # oder 9 | Ziffer optional | Leerzeichen | Jede Ziffer ist erlaubt. |
| A | Buchstabe | Leerzeichen | Jeder Buchstabe [a-zA-Z] als ASCII-Zeichen ist erlaubt. |
| < | rechts-ausgerichtetes Feld | - | Die Zeichen zwischen dem letzten Separator und < sind rechtsbündig ausgerichtet. |
| ! | Fokus | - | Definiert das Zeichen, hinter dem der Cursor steht, wenn die MaskBox den Fokus erhält. |
| Alle anderen Zeichen | Separator | - | Jedes andere Zeichen ist ein Trennzeichen. |
| \ | Quoting | - | Wenn Sie ein spezielles Zeichen als Separator einsetzen, dann müssen Sie dieses Zeichen maskieren. |

Tabelle 16.7.1 Übersicht zu den Zeichen in einer MaskBox

Beispiele für ausgewählte Masken:

- Geldwert → "#####0<!,00 €"
- Datum → "99.99.9999"
- IPv4 → "##0<.##0<.##0<.##0<"
- Farbwert (hex) → "&H![0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F]&"
- Bildgröße → "[0-9][0-9][0-9]x[0-9][0-9][0-9] [px]"

Die Vorgabe einer Datum-Maske zum Beispiel impliziert nicht notwendigerweise, dass die Eingabe des Datums 29.02.2013 – im Gegensatz zu 29.02.2012 – als fehlerhaftes Datum erkannt wird. Sie müssen Eingaben prüfen, die aus einer MaskBox ausgelesen werden, um valide Daten im Programm zu verwenden. Somit kommen zwei Aufgaben auf Sie zu, wenn Sie die Komponente MaskBox als spezialisierte TextBox einsetzen wollen:

- Wahl und Erprobung einer geeigneten MaskBox-Maske und
- Prüfung der Zeichenkette *MaskBox.Text* mit einem geeigneten Prüfverfahren, um nur mit validen Daten weiterzuarbeiten.

In einem Projekt mit 5 MaskBoxen für unterschiedliche Eingaben wird gezeigt, wie Sie diese beiden Aufgaben erfolgreich bearbeiten. Daher wird der vollständige Quelltext vorgestellt. Sie finden dort die speziellen Eingabe-Masken und die Prüfprozeduren, die alle auf die Komponente *gb.pcre* setzen.

```
' Gambas class file

Public sMaske As String
Private sSubject As String
Private sPattern As String

Public Sub Form_Open()
  FMain.Center()
  FMain.Resizable = False
```

```

mboxTime.Mask = "00:00 Uhr"
mboxCurrency.Mask = "#####0<! ,00 €"
mboxColor.Mask = "&H![0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F]&"
mboxPictureSize.Mask = "### x ### px" ' Alternative
mboxPictureSize.Mask = "[0-9][0-9][0-9]x[0-9][0-9][0-9] ]px" ' 999 x 100
mboxPoint3D.Mask = "P(!#0|#0|#0)" ' Koordinaten im Intervall 00..99

End ' Form_Open()

*****

Public Sub MBGroup_Activate()

Select Last.Tag
Case "T"
    lblValue.Text = MBGroupCheck(mboxTime.Text, "T")
Case "G"
    lblValue.Text = MBGroupCheck(mboxCurrency.Text, "G")
Case "F"
    lblValue.Text = MBGroupCheck(mboxColor.Text, "F")
Case "B" ' Default
    lblValue.Text = MBGroupCheck(mboxPictureSize.Text, "B")
Case "3D" ' Default
    lblValue.Text = MBGroupCheck(mboxPoint3D.Text, "3D")
End Select

End ' MBGroup_Activate()

Public Sub MBGroup_DblClick()
    MBGroup_Activate()
End ' MBGroup_DblClick()

Public Function MBGroupCheck(sMBText As String, sTag As Variant) As String
    Dim aMatrix As String[]

Select sTag
Case "T"
    sSubject = Replace(sMBText, " Uhr", "")
    ' Stunden gibt es von 00-19 Uhr und von 20-23 und 24 Uhr ist 00:00 Uhr
    ' Minuten von 00 bis 59
    sPattern = "^([01][0-9]|2[0-3]):[0-5][0-9]$" ' hh:mm
    ' sPattern = "^([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9]$" ' hh:mm:ss mit Sekunden
    If Match(sSubject, sPattern) = True Then
        Return "Zeit-Eingabe O.K."
    Else
        Return "Zeit-Eingabe falsch!"
    Endif ' Match(...) = True ?
Case "G"
    sSubject = sMBText
    sPattern = "^([ 0-9]+,[0-9]{2})( €)$"
    If Match(sSubject, sPattern) = True Then
        Return "Geldwert-Eingabe O.K."
    Else
        Return "Geldwert-Eingabe falsch!"
    Endif ' Match(...) = True ?
Case "F"
    sMBText = Replace(sMBText, "&H", "")
    sMBText = Replace(sMBText, "&", "")
    sSubject = sMBText
    sPattern = "^([0-9a-fA-F]{6})$"
    If Match(sSubject, sPattern) = True Then
        mboxColor.Text = "&H" & Upper(sMBText) & "&"
        Return "FarbWert-Eingabe (hex) O.K."
    Else
        Return "Farbwert-Eingabe falsch!"
    Endif ' Match(...) = True ?
Case "B"
    sSubject = Replace(sMBText, " px", "")
    sPattern = "^([0-9]{3})x([0-9]{3})$"
    If Match(sSubject, sPattern) = True Then
        Return "Bildgrößenangabe O.K."
    Else
        Return "Bildgrößenangabe falsch!"
    Endif ' Match(...) = True ?
Case "3D"
    sMBText = Replace(sMBText, "P(", "")
    sMBText = Replace(sMBText, ")", "")
    aMatrix = Split(sMBText, "|")
    Return "x = " & aMatrix[0] & ", y = " & aMatrix[1] & ", z = " & aMatrix[2]
End Select

End ' MBGroupCheck(...)

```

```
Public Function Match(Subject As String, Pattern As String) As Boolean
    Dim rRegex As Regexp
    ' Die Komponente gb.pcre (Perl Compatible Regular Expression) muss eingebunden sein
    rRegex = New Regexp(Subject, Pattern)
    If rRegex.Offset = -1 Then
        Return False
    Else
        Return True
    Endif ' rRegex.Offset = -1
End ' Match(...) As Boolean
```

Die Verwendung der Group- und Tag-Eigenschaft für die 5 Maskboxen ist nicht notwendig, wenn Sie in Ihrem Projekt nur eine Prüfung vornehmen. Für die Koordinaten eines Punktes im Raum entfällt durch die Wahl der Maske eine Prüfung, weil nur zulässige Koordinaten eingegeben werden können. Das trifft für die Maske `mboxPoint3D.Mask = "P(!##|##|###)"` jedoch nicht zu – hier muss geprüft werden! Es werden auch mit dieser Maske für jede Koordinate 2 (Ziffern-)Stellen reserviert, der Default-Wert für # ist aber ein Leerzeichen und keine Null wie beim Maskenzeichen 0.

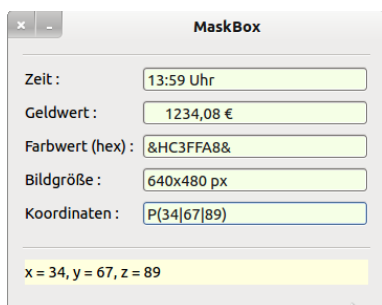


Abbildung 16.7.1: Projekt MaskBox

Nach der Eingabe in einer MaskBox können Sie mit *Enter* oder einem Doppelklick auf die MaskBox die Eingabe-Zeichenkette prüfen und sich das Ergebnis anzeigen lassen.