12.4.1 Klasse Dialog (gb.form.dialog)

Die Komponente definiert erweiterte Versionen der Standard-Dialoge, basierend auf den in der Komponente *gb.qt4* definierten Standard-Dialogen zum Aufruf von Dialog-Boxen. In diesem Kapitel werden Eigenschaften und Methoden der Klasse Dialog (gb.form.dialog) vorgestellt und diese durch Beispiele ergänzt.

12.4.1.1 Eigenschaften

Die Klasse *Dialog (gb.form.dialog)* verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
AutoExt	Boolean	Gibt den Wert zurück oder legt mit *True* fest, dass die Datei-Extension im Filter im Datei-Speichern-Dialog als Vorgabe-Datei-Extension eingestellt wird.
FixedOnly	Boolean	Gibt den Wert zurück oder legt mit *True* fest, dass nur *nicht-proportionale* Schriftarten im Fontauswahl-Dialog anzeigt werden sollen.
ShowHidden	Boolean	Gibt den Wert zurück oder legt mit *True* fest, dass auch versteckte Dateien in der Dialog-Box angezeigt werden sollen.
Color	Integer	Ermittelt die im Farbauswahl-Dialog gewählte Farbe oder legt die Farbe im Farbauswahl-Dialog als Vorgabe-Farbe fest.
Date	Date	Gibt das im Dialog ausgewählte Datum zurück oder setzt das Datum im Kalender als Vorgabe-Datum.
Filter	String[]	Ermittelt den im Dialog verwendeten Filter (Datei-Extension) oder legt den Filter fest. *Filter* ist ein String-Array, bei dem jeder Eintrag aus einem Filter und einer Filter-Beschreibung besteht. Ein Filter ist eine Liste von Datei-Extensionen mit Platzhaltern, die durch ein Semikolon getrennt sind. Eine Filterbeschreibung kann eine beliebige Zeichenfolge sein. Ein Filter wird bei der Anzeige automatisch an die Filterbeschreibung angehängt.
FilterIndex	Integer	Durch Setzen des Arrays-Index kann man die Vorauswahl des Filters verändern, der beim Öffnen des Dialogs angezeigt wird (Vorgabe = 0). Durch Abfrage des Index kann ermittelt werden, welcher Filter ausgewählt wurde. Verfügbar in Gambas-Versionen > 3.16.3 .
Font	Font	Ermittelt den im Fontauswahl-Dialog gewählten Font oder legt den Font im Fontauswahl-Dialog als Vorgabe-Font fest.
Path	String	Gibt den ausgewählten Datei-Pfad im Datei-Auswahl-Dialog als String zurück oder legt den Datei-Pfad im Datei-Dialog als Vorgabe-Pfad fest.
Paths	String[]	Gibt die ausgewählten Datei-Pfade im Datei-Öffnen-Dialog als String-Array zurück.
Title	String	Ermittelt den im Dialog gewählten (Fenster-)Titel oder legt den Titel für den Dialog fest.

Tabelle 12.4.1.1.1 : Eigenschaften der Klasse Dialog (gb.form.dialog)

12.4.1.2 Methoden

Klasse *Dialog (gb.form.dialog)* besitzt die folgenden Methoden. Beachten Sie optionale Argumente.

Methode	Beschreibung
OpenFile([Multi As Boolean]) As Boolean	Ruft den Datei-Öffnen-Dialog auf, um den Datei-Namen der zu öffnenden Datei abzufragen. Wenn das optionale Argument 'Multi' den Wert False (Standard) hat, so kann der Benutzer genau eine Datei auswählen. Der Rückgabewert ist der Pfad zur ausgewählten Datei - gespeichert in der Pfad-Eigenschaft Dialog.Path. Wenn das *optionale* Argument 'Multi' den Wert *True* hat, so kann der Benutzer mehrere Dateien auswählen. Der Rückgabewert ist ein String-Array, das die Pfade zu allen ausgewählten Dateien enthält - gespeichert in der Eigenschaft Dialog.Paths. Die Methode selbst gibt *True* zurück, wenn der Benutzer den Abbruch-Button gedrückt hat oder *False*, wenn der Benutzer auf den OK-Button gedrückt hat.

Methode	Beschreibung
SaveFile( ) As Boolean	Ruft den (Standard-)Datei-Speichern-Unter-Dialog auf, um den Datei-Namen der zu speichernden Datei abzufragen. Die Methode selbst gibt *True* zurück, wenn der Benutzer den Abbruch-Button gedrückt hat oder *False*, wenn der Benutzer auf den OK-Button gedrückt hat.
SelectDirectory( ) As Boolean	Ruft den Datei-Standarddialog auf, um einen *existierenden* Verzeichnisnamen auszulesen. Die Methode selbst gibt *True* zurück, wenn der Benutzer den Abbruch-Button gedrückt hat oder *False*, wenn der Benutzer auf den OK-Button gedrückt hat.
SelectFont( ) As Boolean	Ruft den Standard-Fontauswahl-Dialog auf. Die Methode selbst gibt *True* zurück, wenn der Benutzer den Abbruch-Button gedrückt hat oder *False*, wenn der Benutzer auf den OK-Button gedrückt hat.
SelectColor() As Boolean	Ruft den Standard-Farbauswahl-Dialog auf. Die Methode selbst gibt *True* zurück, wenn der Benutzer den Abbruch-Button gedrückt hat oder *False*, wenn der Benutzer auf den OK-Button gedrückt hat.
SelectDate( ) As Boolean	Ruft das Steuerelement 'DateChooser' auf, um ein Datum auszuwählen. Die Methode selbst gibt *True* zurück, wenn der Benutzer den Abbruch-Button gedrückt hat oder *False*, wenn der Benutzer auf den OK-Button gedrückt hat.

Tabelle 12.4.1.2.1 : Methoden der Klasse Dialog (gb.form.dialog)

12.4.1.3 Beispiele

Um alle Beispiele nachvollziehen zu können, finden Sie im Download-Bereich ein Projekt-Archiv. Aus diesem Grund wird hier auf den kompletten Quelltext verzichtet. Es werden nur Quelltext-Ausschnitte vorgestellt, in denen wichtige Passagen farbig hervorgehoben werden und die erzielten Ergebnisse angezeigt.

Beispiel 1 – Dialog.SelectDirectory()

Im ersten Beispiel wird ein (existierendes) Verzeichnis ausgewählt:

```
Public Sub btnSelectDirectory_Click()
  Dialog.Title = "Wählen Sie ein Bild-Verzeichnis aus ..."
  Dialog.Path = Application.Path
  If Dialog.SelectDirectory() Then Return

  sImageDirectoryPath = Dialog.Path
  btnOpenFileImage.Enabled = True
End ' btnSetFont_Click()
```

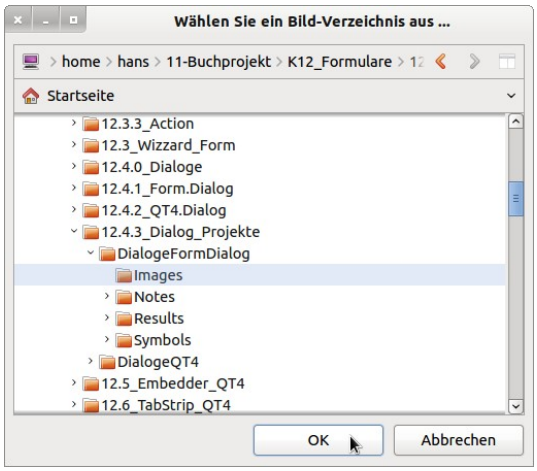

Abbildung 12.4.1.3.1: Verzeichnis-Auswahl-Dialog-Box

Beispiel 2 – Dialog.Openfile(..) – Bilddateien

Aus dem im ersten Beispiel ausgewählten Verzeichnis soll genau eine bestimmte (Bild-)Datei ausge-

wählt werden – das optionale Argument hat den Wert False (Standard). Neben dem Datei-Filter wird festgelegt, dass versteckte Dateien nicht angezeigt werden:

```
Public Sub btnOpenFileImage_Click()
  Dialog.Title = "Wählen Sie eine Bild-Datei aus ..."
' Dialog.Filter = ["*.jpg", "JPG-Bilddatei", "*.png", "PNG-Bilddatei", "*", "Alle Dateien"]
  Dialog.Filter = ["*.png;*.jpg;*.jpeg;*.gif", "Bild-Dateien", "*", "Alle Dateien"]
  Dialog.ShowHidden = False
  If Not sImageDirectoryPath Then Dialog.Path = Application.Path &/ "Images"
  If Dialog.Openfile(False) Then Return ' Genau 1 Datei auswählen (False -> Multiselect ausgeschaltet)

  sImagePath = Dialog.Path ' Pfad sichern
  PictureBoxD.Picture = Picture.Load(Dialog.Path)
  FMain.Text = "Bearbeitet wird die Bild-Datei:  " & File.Name(Dialog.Path)
  btnOpenFileText.Enabled = True
  Catch
    Message.Info(Error.Text)
End ' btnOpenFileImage_Click()
```

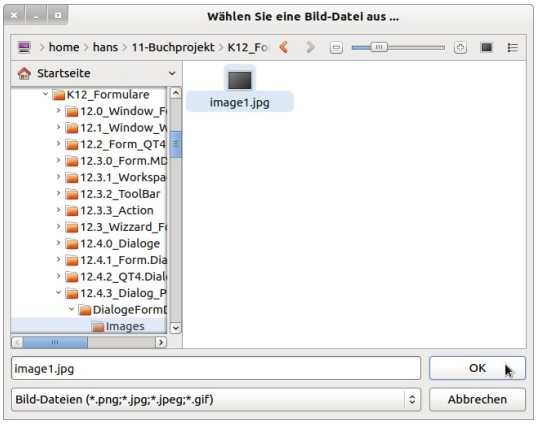

Abbildung 12.4.1.3.2: Datei-Öffnen-Dialog-Box – 1

Der erste Filter wird nicht verwendet, sondern das Aktivierte als kompakte Filterliste. Beim ersten Filter werden jeweils die *einzelnen* Filter untereinander in der Combo-Box angezeigt.

Beispiel 3 – Dialog.Openfile(..) – Textdateien

Es wird genau eine Text-Datei mit der Extension *txt* ausgewählt. In diesem Projekt besitzen die einzelnen Zeilen ein bestimmtes Format.

```
Public Sub btnOpenFileText_Click()
  Dim i As Integer
  Dim aTextArray, aBildtext As String[]

  Dialog.Title = "Wählen Sie eine Bildbeschreibungsdatei aus ..."
  Dialog.Filter = ["*.txt", "Text-Dateien", "*", "Alle Dateien"]
  Dialog.Path = Application.Path &/ "Images"
  If Dialog.OpenFile(False) Then Return ' Genau 1 Datei auswählen (False -> Multiselect ausgeschaltet)

  aTextArray = Split(File.Load(Dialog.Path), gb.NewLine) ' Jede Text-Zeile ist ein Element im Array
  aBildtext = New String[2] ' Weiteres Array mit 2 Elementen erzeugen
  For i = 0 To 1
  ' Nur die ersten beiden Zeilen werden am Trennzeichen ':' zerlegt und jeweils der 2. Teil gespeichert
    aBildtext[i] = Split(aTextArray[i], ":")[1]
  Next

  lblBildtext1.Text = Trim(aBildtext[0])
  lblBildtext2.Text = Trim(aBildtext[1])
  btnSelectFont.Enabled = True
  Catch
    Message.Info(Error.Text)
End ' btnOpenFileText_Click()
```

So sehen die ersten Zeilen in der Text-Datei aus:

```
Bildtitel: Flora der Alpen
Bildtitel2: Augentrost (Euphrasia minima) - Allgäu - 2012
Kameramodell: COOLPIX L22
Name: image1.jpg
Breite: 510 Pixel
```

```
Höhe: 294 Pixel
Typ: JPEG-Bild
Bytes: 42,6 kB
Blendenwert: f/5,5
...
```

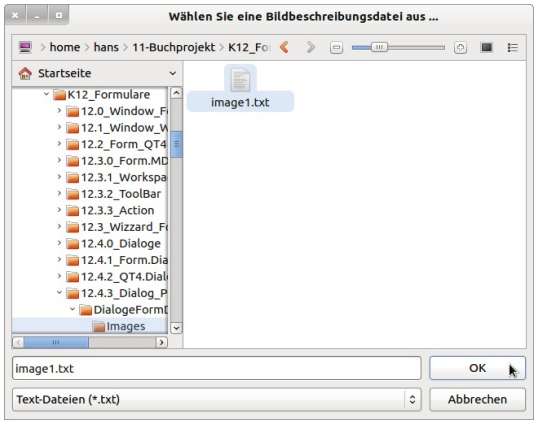

Abbildung 12.4.1.3.3: Datei-Öffnen-Dialog-Box – 2

Beispiel 4 – Dialog.SelectFont()

Neben dem Titel der Dialog-Box wird auch der Font eines bestimmten Label-Steuerelements als Vorgabe-Font festgelegt. Zusätzlich wird vereinbart, dass *alle* existierenden Fonts zur Auswahl stehen und nicht nur die nichtproportionale Schriftarten angezeigt werden. Auf diese Vereinbarung können Sie auch verzichten, da als Standard 'Dialog.FixedOnly = False' gilt.

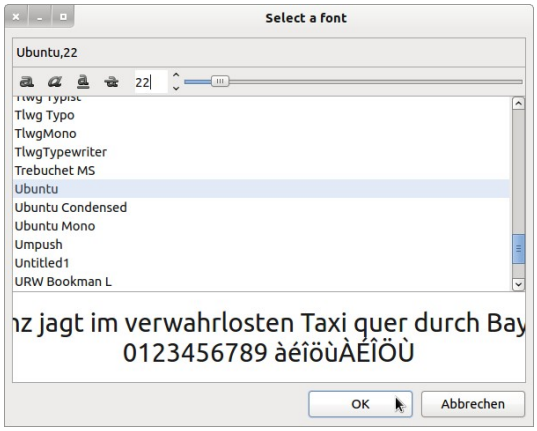

Abbildung 12.4.1.3.4: Font-Dialog-Box

```
Public Sub btnSelectFont_Click()
  Dialog.Title = "Wählen Sie eine Schriftart aus ..."
  Dialog.Font = lblBildtext1.Font
  Dialog.FixedOnly = False
  If Dialog.SelectFont() Then Return

  lblBildtext1.Font = Dialog.Font
  lblBildtext1.Font.Size = 0.7 * Dialog.Font.Size
  lblBildtext2.Font = Dialog.Font
  lblBildtext2.Font.Size = 0.7 * 0.8 * Dialog.Font.Size
  lblBildtext2.Font.Bold = False
  hFont = Dialog.Font ' Aktuellen Font sichern
  lblBildtext2.Tooltip = lblBildtext2.Text
  btnSelectColor.Enabled = True
End ' btnSelectFont_Click()
```

Beispiel 5 – Dialog.SelectColor()

Um Farbe in das Spiel zu bekommen, können Sie den Color-Dialog aufrufen. Zusätzlich zum Dialog-Titel wird die Farbe *weiß* als Vorgabe-Farbe deklariert.

```
Public Sub btnSelectColor_Click()
  Dialog.Title = "Wählen Sie eine Farbe aus ..."
  Dialog.Color = Color.White
  If Dialog.SelectColor() Then Return
  lblBildtext1.Foreground = Dialog.Color
  lblBildtext2.Foreground = Color.White
  hColor = Dialog.Color ' Aktuelle Farbe sichern
  btnPreview.Enabled = True
End ' btnSelectColor_Click()
```

Abbildung 12.4.1.3.5: Color-Dialog-Box

Beispiel 6 – Dialog.SaveFile()

Neben der Festlegung des Dialog-Box-Titels, des Vorgabe-Pfades und des Datei-Filters wird zusätzlich die nur im *SaveFile-Dialog* vorhandene Eigenschaft *Dialog.AutoExt* auf den Wert *True* gesetzt. Das bewirkt, dass Sie den Dateinamen der zu speichernden Datei ohne Extension eingeben können; diese wird automatisch ergänzt → Abbildung 12.4.1.3.6 unten links.

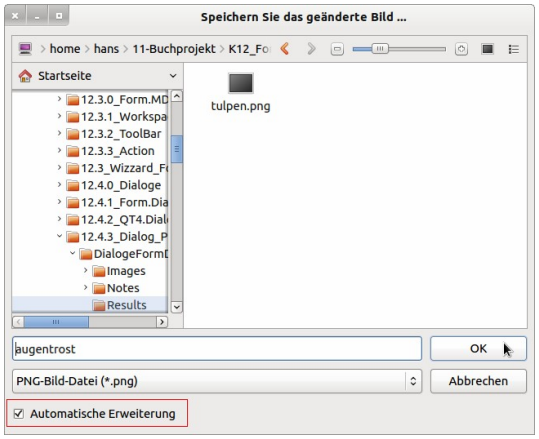

Abbildung 12.4.1.3.6: Datei-Speichern-Dialog-Box

```
Public Sub btnSaveFileImage_Click()
  Dialog.Title = "Speichern Sie das geänderte Bild ..."
  Dialog.Filter = ["*.png", "PNG-Bild-Datei", "*", "Alle Dateien"]
  Dialog.Path = Application.Path &/ "Results/"
  Dialog.AutoExt = True
  If Dialog.SaveFile() Then Return
' File.Save(Dialog.Path, PictureBoxD.Picture) ' Nicht zulässig - es werden nur Strings gespeichert!
  PictureBoxD.Picture.Save(Dialog.Path, 100) ' hImage.Save(Dialog.Path, 100) ' → Alternative
  GetReset()
  Catch
    Message.Info(Error.Text)
End ' btnSaveFileImage_Click()
```

Beispiel 7 – Dialog.Date()

Dieser Dialog zur Auswahl eines Datums stellt insofern eine Besonderheit dar, da der in der Komponente *gb.form* existierende *DateChooser* aufgerufen wird. Sie können auch hier wieder den Dialog-Titel vorgeben und ein Datum; im Beispiel eingestellt auf das aktuelle Datum:

```
Public Sub btnSelectDate_Click()
  Dialog.Title = "Wählen Sie ein Datum aus ..."
```

```
  Dialog.Date = Now()
  If Dialog.SelectDate() Then Return
  dDate = Dialog.Date ' Ausgewähltes Datum sichern
  ' Print Format(dDate, "dddd - dd. mmmm yyyy")
End ' btnSelectDate_Click()
```

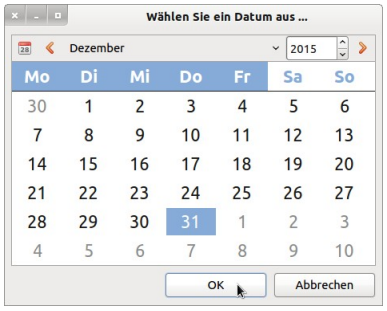

Abbildung 12.4.1.3.7: Dialog → DateChooser

Die oben verwendeten Quelltexte wurden aus einem Projekt entnommen, das Ihnen im Download-Bereich für eigene Experimente zur Verfügung gestellt wird:

Abbildung 12.4.1.3.8: Programm-Oberfläche