

23.4.3 Klasse Picture

Diese Klasse (gb.qt4) repräsentiert ein Bild. Die Daten eines Picture-Objekts (in gb.qt4) werden nicht im Adressraum des Gambas-Prozess gespeichert, sondern liegen im Adressraum des X-Servers. Im Gegensatz dazu hat ein Image-Objekt seinen Inhalt im Adressraum des Gambas-Prozesses. Weitere Informationen können Sie in der QT-Dokumentation nachzulesen unter:

<http://sourceforge.net/p/gambas/code/HEAD/tree/gambas/trunk/comp/src/gb.gui.base/.src/Picture.class>.

Die Klasse *Picture* verhält sich wie ein (statisches) Array (`Picture[]`), das man auslesen und beschreiben kann.

```
Dim hPicture As Picture
hPicture = Picture [ Path As String ]
```

Die zweite Anweisung gibt ein Picture-Objekt aus dem internen Bild-Cache zurück. Wenn das Picture nicht im Cache vorhanden ist, wird es automatisch aus der angegebenen Datei geladen. Beginnt der Pfad mit "icon: /", dann bedeutet es, dass ein Stock-Icon geladen werden soll. Um auf diese Symbole zuzugreifen nutzen Sie die Klasse *Stock* (→ Kapitel 20.8). So fügen Sie ein Picture als Symbol in einen Button ein. Unterstützt werden die Bild-Formate: JPEG, PNG, BMP, GIF, SVG, ICO und XPM:

```
' Lesen
btnClose.Picture = Picture["icon:/16/close"]
```

Mit dieser Anweisung:

```
' Schreiben
Dim hPicture As Picture
...
Picture[Path As String] = hPicture
```

fügen Sie ein Picture in den internen Picture-Cache ein. Es handelt sich bei dem Parameter *Path* nur um einen Schlüssel, der *intern* für eine Collection verwendet wird. Man sollte aber beachten, dass der Schlüssel-Präfix "icon:/" für Gambas-Icons reserviert ist.

23.4.3.1 Ein neues Picture erzeugen

```
Dim hPicture As Picture
hPicture = New Picture ( [ Width As Integer, Height As Integer, Transparent As Boolean ] )
```

Es wird ein neues Picture erzeugt. Wenn die beiden optionalen Parameter *Width* und *Height* nicht angegeben werden, dann existiert zwar ein neues Picture – es ist aber leer! Sie können mit dem optionalen Parameter *Transparent* festlegen, ob das Picture über einen Alpha-Kanal verfügt oder nicht. Der Standard-Wert ist `False`.

In den folgenden beiden Kapiteln werden Eigenschaften und Methoden der Klasse *Picture* vorgestellt.

23.4.3.2 Statische Methoden

Die Klasse *Picture* besitzt zwei statische Methoden: `Picture.Load` und `Picture.Flush`.

```
Static Function Load ( Path As String ) As Picture → Lädt ein Picture vom Datenträger.
Static Sub Flush ( ) → Leert den internen Picture-Cache.
```

23.4.3.3 Eigenschaften

Die Klasse *Picture* verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Depth	Integer	Gibt die Farbtiefe eines Pictures zurück. Der zurückgegebene Wert ist abhängig vom X11-Server. Er sollte 32 Bits sein.
Height	Integer	Gibt die Höhe eines Pictures zurück.
H	Integer	Synonym für Height

Eigenschaft	Datentyp	Beschreibung
Width	Integer	Gibt die Breite eines Pictures zurück.
W	Integer	Synonym für Width
Image	Image	Konvertiert ein Picture in ein Image und gibt das Image-Objekt zurück.
Transparent	Boolean	Gibt an, ob das Picture über einen Alpha-Kanal verfügt oder nicht oder setzt diesen Wert.

Tabelle 23.4.3.3.1 : Eigenschaften der Klasse Picture

23.4.3.4 Methoden

Die Klasse *Picture* besitzt die folgenden Methoden. Beachten Sie die optionalen Parameter:

Methode	Beschreibung
Clear ()	Leert das Picture (H=0 und W=0).
Copy ([X As Integer, Y As Integer, Width As Integer, Height As Integer])	Gibt eine 1:1-Kopie eines Pictures zurück oder die Kopie eines Teiles des originalen Pictures, wenn die 4 optionalen Parameter gesetzt werden.
Fill (Color As Integer)	Füllt das Picture mit der angegebenen Farbe.
Resize (Width As Integer, Height As Integer)	Ändert die Größe eines Pictures. Beachten Sie, dass das Picture nicht maßstabsgetreu vergrößert wird, sondern abgeschnitten wird.
Save (Path As String [, Quality As Integer])	Speichert ein Picture auf einen Datenträger unter dem angegebenen (Bild-)Dateipfad. Der optionale Parameter bestimmt ausschließlich für das jpg- beziehungsweise png-Format die Bild-Qualität (0..100) oder die Kompression des abgespeicherten Pictures.

Tabelle 23.4.3.4.1 : Methoden der Klasse Picture

Im Gegensatz zum JPG-Dateiformat ist die Daten-Kompression beim PNG-Format verlustfrei. Der Parameter 'Quality' ist deshalb für den Fall des PNG-Dateiformats erklärungsbedürftig. Mit einem eigens hierfür geschriebenen Gambas-Programm und dem Analyse-Tool 'pngcheck' wurden folgende Ergebnisse ermittelt, wobei die Aufteilung der Bereiche immer dann erfolgte, wenn sich der Kompressionstyp oder die erreichte Kompression änderte.

Als Vorlage diente ein eingescanntes Image einer typischen gedruckten A4-Briefseite:

Qualität	PNG-Kompression-Typ	Erreichte Kompression
0 - 19	maximum compression	91,7 %
20 - 29	maximum compression	91,6 %
30 - 39	default compression	91,6 %
40 - 49	fast compression	91,6 %
50 - 59	fast compression	91,5 %
60 - 69	fast compression	91,2 %
70 - 79	fast compression	91,1 %
80 - 89	superfast compression	90,9 %
90 - 100	superfast compression	- 0,2 %

Tabelle 23.4.3.4.2 : Übersicht Qualitätswert ↔ Kompression

Bitte beachten Sie, dass der Berechnungsaufwand mit sinkendem Qualitätswert spürbar steigt. In der täglichen Praxis hat sich für das PNG-Format ein Qualitätswert von 30 (= Standard-Kompression) bewährt.

Dagegen erzielt man beim verlustbehafteten jpg-Format mit einem Qualitätswert von 90 einen guten Kompromiss zwischen erreichter Bildqualität und Größe der Datei.

23.4.3.5 Projekt

Im Projekt werden alle Eigenschaften und Methoden eingesetzt. Für alle Programm-Teile werden die relevanten Quelltext-Ausschnitte vorgestellt und die wichtigen Passagen farbig hervorgehoben. Die erzielten Ergebnisse werden angezeigt.

Picture erzeugen

```
[1] ' Gambas class file
[2]
[3] Public hPicture1 As Picture
[4]
[5] Public Sub Form_Open()
[6]
[7]     FMain.Center
[8]     FMain.Resizable = False
[9]     PictureBox1.Stretch = False
[10]
[11]     PictureBox1.Picture = Picture["Symbols/intro.jpg"]
[12]     btnClearPic.Picture = Picture["icon:/16/clear"]
[13]     Picture.Flush()
[14]
[15] End
[16]
[17] Public Sub btnGeneratePic_Click()
[18]
[19]     TextArea1.Clear
[20]     hPicture1 = New Picture(PictureBox1.W, PictureBox1.H, True)
[21]     hPicture1.Fill(ColorButton1.Value)
[22]     PictureBox1.Picture = hPicture1
[23]     ShowPropertiesPic(hPicture1)
[24]
[25] End
```

Kommentar:

- In den Zeilen 11 und 12 werden die Bilder aus dem angegebenen Pfad ausgelesen, im Picture-Cache gespeichert und auf einem Button sowie in einer PictureBox *angezeigt*.
- Die verwendete PictureBox hat eine feste Weite von 360·px und eine Breite von 240·px.
- Anschließend wird der Cache geleert.
- Ein neues Picture-Objekt wird in der Zeile 20 mit vorgegebener Größe und angegebener Transparenz-Eigenschaft erzeugt.
- Danach wird das Picture (Zeile 21) in der frei wählbaren Farbe eingefärbt und in der PictureBox angezeigt:



Abbildung 23.4.3.5.1: Ein neues Picture (einfarbig) wird angezeigt

Picture aus Bild-Datei erzeugen



Abbildung 23.4.3.5.2: Anzeige Picture

Das Bild (Breite 640 Pixel, Höhe 390 Pixel) wird nur unvollständig (linke, obere Ecke) in einer Picture-Box angezeigt, weil diese eine feste Größe hat, die hier *kleiner* als das Picture ist.

```
[1] Public Sub btnPictureFromFile_Click()
[2]     TextArea1.Clear
[3]     hPicture1 = Picture.Load("Images/b1.png")
[4]     PictureBox1.Picture = hPicture1
[5]     ShowPropertiesPic(hPicture1)
[6] End
```

Kommentar:

- In den Zeilen 3 und 4 wird ein Bild aus dem angegebenen Pfad ausgelesen, im Picture-Cache gespeichert und dem Picture *hPicture1* zugewiesen.
- Danach wird das Picture (Zeile 17) in der PictureBox angezeigt.

Picture im korrekten Seitenverhältnis vollständig in einer PictureBox anzeigen

Den Nachteil der Anzeige aus dem letzten Bild können Sie beseitigen. Wenn das Picture maßstabsgetreu vergrößert oder verkleinert werden soll, dann müssen Sie das *Picture* zuerst in ein *Image* konvertieren, dieses dann in der Größe ändern und abschließend das *Image* wieder in ein *Picture* konvertieren. Eine *Stretch*-Methode existiert für ein *Picture* nicht!

```
[1] Public Sub btnStretch_Click()
[2]     DoStretch("Images/b1.png")
[3] End
[4]
[5] Private Sub DoStretch(sPath As String)
[6]     Dim transImage As Image
[7]     Dim fRatio, fStretch As Float
[8]
[9]     TextArea1.Clear
[10]    hPicture1 = Picture.Load(sPath)
[11]
[12]    transImage = hPicture1.Image
[13]    If hPicture1.W > hPicture1.H Then
[14]        fRatio = hPicture1.W / hPicture1.H
[15]        PictureBox1.Picture = transImage.Stretch(PictureBox1.W, PictureBox1.W / fRatio).Picture
[16]        fStretch = PictureBox1.W / hPicture1.W
[17]    Else
[18]        fRatio = hPicture1.H / hPicture1.W
[19]        PictureBox1.Picture = transImage.Stretch(PictureBox1.H / fRatio, PictureBox1.H).Picture
[20]        fStretch = PictureBox1.H / hPicture1.H
[21]    Endif
[22]
```

```

[23] TextArea1.Clear
[24] TextArea1.Insert("Picture-Original-Breite = " & hPicture1.W & "px" & gb.NewLine)
[25] TextArea1.Insert("Picture-Original-Höhe = " & hPicture1.H & "px" & gb.NewLine)
[26] TextArea1.Insert("Stretch-Faktor = " & Str(Round(100 * (fStretch), -1)) & "%" & gb.NewLine)
[27] TextArea1.Insert("Picture mit Alpha-Kanal? " & " → " & IIf(hPicture1.Transparent, "Ja", "Nein"))
[28]
[29] End

```



Abbildung 23.4.3.5.3: Maßstabsgetreue Anzeige des Pictures

Kommentar:

- In der Zeile 10 wird dem Picture der Inhalt der Datei *b1.png* zugewiesen.
- Dann wird das Picture in der Zeile 10 in der Größe einerseits maßstabsgetreu geändert und auf die Breite der PictureBox normiert, da im ersten Fall (Zeile 13) die Breite größer als die Höhe des Pictures ist. Andererseits wird das geänderte Image sofort wieder in ein Picture konvertiert und in der PictureBox angezeigt.

Picture-Kopie

Sie können eine 1:1-Kopie eines Pictures erzeugen oder die Kopie eines Bildausschnitts, wenn Sie die 4 optionalen Parameter mit *korrekten* Argumenten setzen. Im Beispiel wird die Kopie eines Bildausschnitts erzeugt und die Breite sowie Höhe so gewählt, dass die Kopie genau in die PictureBox passt:

```

[1] Public Sub btnPartCopyPic_Click()
[2]
[3]     CopyPic(100, 110, PictureBox1.Width, PictureBox1.Height)
[4]
[5] End
[6]
[7] Private Sub CopyPic(iX0 As Integer, iY0 As Integer, iW As Integer, iH As Integer)
[8]     Dim coPicture As Picture
[9]
[10]    TextArea1.Clear()
[11]
[12]    hPicture1 = Picture.Load("Images/b1.png")
[13]    coPicture = hPicture1.Copy(iX0, iY0, iW, iH)
[14]    PictureBox1.Picture = coPicture
[15]
[16]    If coPicture Then
[17]        TextArea1.Clear
[18]        TextArea1.Insert("Xo = " & iX0 & "px " & " | Yo = " & iY0 & "px" & gb.NewLine)
[19]        ...
[20]    Endif
[21]
[22] End

```



Abbildung 23.4.3.5.4: Picture-Kopie als Bildausschnitt

Ein Picture leeren

Ein Picture können Sie mit der Clear-Methode leeren.

```
[1] Private Sub ClearPic(hPic As Picture)
[2]     If hPic Then
[3]         hPic.Clear
[4]         PictureBox1.Picture = hPic
[5]         ShowPropertiesPic(hPic)
[6]     Endif
[7] End
```

Kommentar:

- Ein leeres Picture existiert noch als Picture-Objekt und hat eine Breite von 0 Pixel und eine Höhe von 0 Pixel.
- Ein leeres Picture können Sie nicht in einer Datei speichern.

Ein Picture in einer Datei speichern

Im Projekt wird zum Programm-Ende stets das aktuelle Picture in einer Datei in einem Ordner innerhalb des Projekt-Ordners gespeichert:

```
[1] Public Sub Form_Close()
[2]     If hPicture1 Then
[3]         If hPicture1.W * hPicture1.H <> 0 Then
[4]             hPicture1.Save("Backup/backup.png", 100)
[5]         Endif
[6]     Endif
[7] End
```

Kommentar:

- In der Zeile 2 wird geprüft, ob das Picture *hPicture1* existiert.
- Um ein leeres Picture auszuschließen, wird in der Zeile 3 geprüft, ob das Produkt aus Länge und Breite des Pictures einen Wert ungleich 0 ergibt. Null ergäbe sich genau dann, wenn mindestens ein Faktor 0 ist.

Mit 'Stretch 2' und 'Stretch 3' werden Pictures angezeigt, bei denen die Höhe größer als die Breite ist oder beide Werte gleich sind. Für die Anzeige werden die Höhen auf die Höhe der PictureBox normiert:

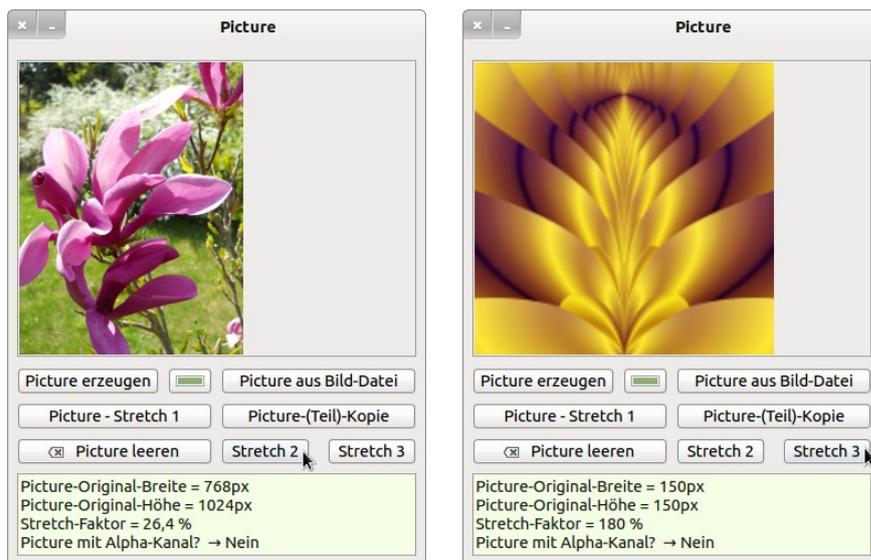


Abbildung 23.4.3.5.5: Maßstabsgetreue Anzeige von Pictures

Für die beiden Button 'Stretch 2' und 'Stretch 3' existieren ToolTips mit folgendem Inhalt:

Mit 'Stretch 2' wird ein Picture angezeigt, bei dem die Höhe größer als die Breite ist. Für die Anzeige wird die Picture-Höhe auf die Höhe der PictureBox normiert.

Mit 'Stretch 3' wird ein Picture angezeigt, bei dem Höhe und Breite gleich sind. Für die Anzeige wird die Picture-Höhe auf die Höhe der PictureBox normiert.