

### 23.12.3 Klasse PdfPage (gb.poppler)

Die Klasse PdfPage ist virtuell und kann nicht erzeugt werden. Diese virtuelle Klasse repräsentiert eine Seite eines PDF-Dokuments. So greifen Sie in einem PDF-Dokument auf eine bestimmte Seite über ihren Seiten-Index zu:

```
Dim iIndex As Integer

$hPdfDocument = New PdfDocument(Dialog.Path)

If $hPdfDocument.Count > 0 Then
  For iIndex = 0 To $hPdfDocument.Max
    Print "Label = "; $hPdfDocument[iIndex].Label
    Print "Height = "; $hPdfDocument[iIndex].H
    Print "Weight = "; $hPdfDocument[iIndex].W
    Print "Text = "; $hPdfDocument[iIndex].Text
  Next
EndIf
```

#### 23.12.3.1 Eigenschaften

Die virtuelle Klasse PdfPage verfügt über die folgenden Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Height oder H	Integer	Gibt die Seitenhöhe einer PDF-Seite in Pixeln zurück.
Width oder W	Integer	Gibt die Seitenbreite einer PDF-Seite in Pixeln zurück.
Label	String	Gibt das Label (Seitenzahl) einer PDF-Seite zurück.
Text	String	Gibt den Text <u>einer</u> PDF-Seite zurück.
Thumbnail	Image	Gibt das Vorschaubild einer PDF-Seite zurück. Der Rückgabewert ist Null, wenn <u>kein</u> Miniaturbild der PDF-Seite im PDF-Dokument vorhanden ist oder wenn es in einem nicht unterstützten Bildformat vorliegt.

Tabelle 23.12.3.1.1 : Eigenschaften der virtuellen Klasse PdfPage

##### 23.12.3.1.1 Hinweise zur Eigenschaft Label

Beachten Sie, dass Seitenlabel und Seitenindex nicht notwendigerweise übereinstimmen. Der Grund liegt darin, dass es zum Beispiel mit Adobe Acrobat © möglich ist, die Seitenzahlen so zu ändern, um ein PDF-Dokument mit der Seitenzahl 3 zu beginnen. In einem PDF-Leser wie XReader sähe das Ergebnis dann so aus, dass das PDF-Dokument mit der Seitenzahl 3 anfängt. Offensichtlich hat ein PDF-Dokument tatsächlich Label für die Seitenzahl und die sind beliebig änderbar.

#### 23.12.3.2 Methoden

Die Klasse PdfPage verfügt über diese drei Methoden:

Methode	Rückgabotyp	Beschreibung
FindText ( Search As String [ , Options As Integer ] )	RectF[ ]	Gibt das Rechteck zurück, in dem die gesuchte Zeichenkette steht. 'Search' ist der Suchtext und 'Options' (optional) ist eine Suchoption oder eine Kombination von Suchoptionen (Konstanten) aus der Klasse Pdf.
GetText ( X As Float, Y As Float, Width As Float, Height As Float )	String	Gibt den Text in dem angegebenen Rechteck zurück. Dabei gilt: X ist die x-Koordinate vom Rechteckpunkt links oben, Y ist die y-Koordinate vom Rechteckpunkt links oben, Width ist die Breite des Rechtecks und Height ist die Höhe des Rechtecks.
Render ( [ X As Integer, Y As Integer, Width As Integer, Height As Integer, Rotation As Integer, Resolution As Float ] )	Image	Rendert die Seite und gibt das resultierende Bild zurück. <b>Beachten Sie den Hinweis im Absatz 23.12.3.3</b>

Tabelle 23.12.3.2.1 : Methoden der virtuellen Klasse PdfPage

Für die Demonstration des Einsatzes der Methode FindText ( Search As String [ , Options As Integer ] ) wird angenommen, dass sich in der Datei searchtext.pdf zum Beispiel dieser Text befindet:

```
G A M B A S - I N F O R M A T I O N E N
-----
Er programmiert in der Programmiersprache Gambas.
Viele Konstanten gelten nur gambas-intern.
Seit gestern spricht er nur noch gambasisch ... .
Fazit: Gambas ist toll!
```

Mit dem folgenden Quelltext kann in einer PDF-Datei nach einem bestimmten Text gesucht werden. In der Konsole der IDE werden die Seite, die Anzahl der Fundstellen auf der Seite sowie die Koordinaten des text-umschließenden Rechtecks ausgegeben:

```
Public Sub SearchText(sSearchText As String, Optional iSearchOption As Integer)

    Dim i As Integer
    Dim aRectF As New RectF[]
    Dim iFound As Boolean

    If IsNull(iSearchOption) Then iSearchOption = 0

    For i = 0 To $hPdfDocument.Max
        aRectF = $hPdfDocument[i].FindText(sSearchText, iSearchOption)
        If aRectF.Count > 0 Then
            iFound = True
            Print ("Found on page ") &
                $hPdfDocument[i].Label & (" | Number of places where the word `") &
                sSearchText & ("` was found = "); aRectF.Count
            Print ("with the coordinates:")
            Print String(40, "-")
            For Each hRectF As RectF In aRectF
                Print "x: "; Round(hRectF.X, -1); " y: "; Round(hRectF.Y, -1);
                Print " | w: "; Round(hRectF.W, -1); " h: "; Round(hRectF.H, -1)
            Next
            Endif
        Next

    If Not iFound Then Print ("The word `") & sSearchText & ("` was not found in the PDF file!")

End
```

Aufruf der o.a. Prozedur mit dem Suchtext "Gambas" und zwei verknüpften Suchoptionen:

```
SearchText("Gambas", Pdf.CaseSensitive Or Pdf.WholeWordsOnly)

Gefunden auf Seite 1 | Anzahl der Fundstellen des Wortes `Gambas` = 1
Rechteck-Koordinaten:
-----
x: 66,6 y: 736,3 | w: 52,9 h: 15,6
```

Aufruf der Prozedur (mit interner Standard-Suchoption) und dem Suchtext "Gambas":

```
SearchText("Gambas")

Gefunden auf Seite 1 | Anzahl der Fundstellen des Wortes `Gambas` = 4
Rechteck-Koordinaten:
-----
x: 306,9 y: 784,6 | w: 52,9 h: 15,6
x: 204,1 y: 768,5 | w: 49,8 h: 15,6
x: 226,9 y: 752,4 | w: 49,8 h: 15,6
x: 66,6 y: 736,3 | w: 52,9 h: 15,6
```

Aufruf der Prozedur mit dem Suchtext "Gambass" und einer Suchoption:

```
SearchText("Gambass", Pdf.CaseSensitive)

Das Wort `Gambass` wurde in der PDF-Datei nicht gefunden!
```

Setzen Sie die Methode GetText ( X As Float, Y As Float, Width As Float, Height As Float ) ein, dann können Sie aus einem PDF-Dokument entweder aus jeder Seite den kompletten vorhandenen Text extrahieren oder jeweils aus einem bestimmten Abschnitt im PDF-Dokument:

```

Public Sub GetPlainText()

    Dim i As Integer

    For i = 0 To $hPdfDocument.Max
        Print "::::: PAGE "; i + 1; " :::::::::::::::::::::"
        Print "SEITEN-LABEL = "; $hPdfDocument[i].Label
        If $hPdfDocument[i].Text.Len > 0 Then
            Print $hPdfDocument[i].GetText(0, 0, $hPdfDocument[i].W, $hPdfDocument[i].H)
            '-- Variant:
            '-- Print $hPdfDocument[i].Text & gb.NewLine
        Else
            Print "No text exists!"
        Endif
    Next

End

```

### 23.12.3.3 Änderung Render()-Methode

Benoit Minisini hat die Render(...) -Methode so geändert (ab Version 3.19.1), dass die beste Auflösung zur Darstellung in einer DocumentView automatisch berechnet wird, wenn man die Argumente Breite und Höhe angibt, aber nicht das Argument Auflösung. Auf diese Weise müssen Sie sich nicht mehr mit dem Problem der Umrechnung zwischen Pixeln, Auflösung und absoluter Größe auseinandersetzen! So können Sie den geänderten Quelltext in einem PDF-Projekt einsetzen:

```

Public Sub DocumentView1_Draw(Page As Integer, Width As Integer, Height As Integer)

    Dim hImage As Image

    hImage = $PDF_Doc[Page].Render(0, 0, Width, Height)
    Paint.DrawImage(hImage, 0, 0)

End

```