

# **pbPDF - Modul**

PDF-Dokumente erstellen

Thorsten Hoeppner

## 1. Überblick über die Befehle

### 1.1 Basis-Befehle

PDF::AddPage  
PDF::Close  
PDF::Create  
PDF::Bookmark  
PDF::Cell  
PDF::DividingLine  
PDF::Image  
PDF::Ln  
PDF::MultiCell  
PDF::MultiCellList  
PDF::PlaceText  
PDF::Rotate  
PDF::SubWrite  
PDF::TruncateCell  
PF::Write

### 1.2 Basis-Einstellungen

PDF::SetAutoPageBreak  
PDF::SetColorRGB  
PDF::SetDashedLine  
PDF::SetFont  
PDF::SetFontSize  
PDF::SetInfo  
PDF::SetPageNumbering  
PDF::SetLineThickness  
PDF::SetLocales  
PDF::SetMargin  
PDF::SetPageMargins  
PDF::SetPosXY  
PDF::SetPosX  
PDF::SetPosY

PDF::GetFontSize  
PDF::GetMargin  
PDF::GetMultiCellLines  
PDF::GetPageHeight  
PDF::GetPageNumber  
PDF::GetPageWidth  
PDF::GetPosX  
PDF::GetPosY  
PDF::GetStringWidth

### 1.3 Fortgeschrittene Befehle

PDF::AddFileLink  
PDF::EmbedJavaScript  
PDF::EmbedJavaScriptFile  
PDF::AddLaunchAction  
PDF::AddLinkURL  
PDF::AddPopupAction  
PDF::AddTextNote  
PDF::EmbedFile  
PDF::EmbedFont  
PDF::EnableFooter  
PDF::EnableHeader  
PDF::EnableTOCNums  
PDF::EscapeText  
PDF::ImageMemory  
PDF::InsertTOC  
PDF::LinkArea  
PDF::StartPageNumsTOC  
PDF::StopPageNumsTOC

## 1.4 Fortgeschrittene Einstellungen

PDF::GetErrorCode  
PDF::GetErrorMessage  
PDF::GetNumbering  
PDF::GetObjectNum  
  
PDF::GetScaleFactor  
PDF::GetWordSpacing  
PDF::SetAliasTotalPages  
PDF::SetColorCMYK  
PDF::SetEncryption  
PDF::SetFooterProcedure  
PDF::SetHeaderProcedure  
PDF::SetOpenAction  
PDF::SetPageCompression  
PDF::SetPageLayout  
PDF::SetPageMode  
PDF::SetViewerPreferences  
PDF::SetWordSpacing

## 1.5 Befehle für die Formularerstellung

PDF::ButtonField  
PDF::ChoiceField  
PDF::TextField

## 1.6 Befehle zum Erzeugen geometrischer Formen

PDF::DrawCircle  
PDF::DrawEllipse  
PDF::DrawGrid  
PDF::DrawLine  
PDF::DrawRectangle  
PDF::DrawRoundedRectangle  
PDF::DrawSector  
PDF::DrawTriangle  
PDF::PathBegin  
PDF::PathEnd  
PDF::PathArc  
PDF::PathLine  
PDF::PathRect

## 1.7 Befehle für Transformationen

PDF::StartTransform  
PDF::StopTransform  
PDF::MirrorHorizontal  
PDF::MirrorVertical  
PDF::Scale  
PDF::ScaleHorizontal  
PDF::ScaleVertical  
PDF::SkewHorizontal  
PDF::SkewVertical  
PDF::Translate  
PDF::TranslateHorizontal  
PDF::TranslateVertical

## 2. Basis – Befehle

### PDF::Create

#### Syntax

```
Create(ID.i, Orientation.s, Unit.s, Format.s)
```

#### Beschreibung

Neues PDF-Dokument mit der entsprechenden **ID** erstellen.

**Orientation** [Default: **#Portrait**]

*Standard-Ausrichtung der Seiten des Dokumentes:*

**#Portrait** oder **"P"** ( Hochformat )

**#Landscape** oder **"L"** ( Querformat )

**Unit** [Default: **mm**]

*Benutzermaßeinheit für das Dokument:*

**pt** ( Point )

**mm** ( Millimeter )

**cm** ( Zentimeter )

**in** ( Inch )

**Format** [Default: **#Format\_A4**]

*Standard Papierformat für das Dokument:*

**#Format\_A0** – **#Format\_A10**

**#Format\_B0** – **#Format\_B10**

**#Format\_C0** – **#Format\_C10**

**#Format\_RA0** – **#Format\_RA4**

**#Format\_SRA0** – **#Format\_SRA4**

**#Format\_Letter**

**#Format\_Legal**

**#Format\_Executive**

**#Format\_Folio**

## PDF : :AddPage

### Syntax

AddPage(ID.i, Orientation.s, Format.s)

### Beschreibung

Seite zum Dokument hinzufügen.

**Orientation** [Default: *Standard des Dokuments*]

*Ausrichtung der neuen Seite:*

#Portrait oder "P" (Hochformat)

#Landscape oder "L" (Querformat)

**Format** [Default: *Standard des Dokuments*]

*Papierformat für die neue Seite:*

#Format\_A0 – #Format\_A10

#Format\_B0 – #Format\_B10

#Format\_C0 – #Format\_C10

#Format\_RA0 – #Format\_RA4

#Format\_SRA0 – #Format\_SRA4

#Format\_Letter / #Format\_Legal / #Format\_Executive / #Format\_Folio

## PDF : :Bookmark

### Syntax

BookMark(ID.i, Titel.s, SubLevel.i, Y.f, Page.i)

### Beschreibung

Lesezeichen für aktuelle Seite definieren.

#### Titel

*Titel des Lesezeichens*

**SubLevel** [Default: 0]

*Lesezeichen-Ebene (oberste Ebene = 0)*

**Y** [Default: #PB\_Default]

*Vertikale Position des Lesezeichens auf der Seite. (#PB\_Default = aktuelle Position)*

**Page** [Default: #PB\_Default]

*Seite für die das Lesezeichens gilt. (#PB\_Default = aktuelle Seite)*

## PDF::Cell

### Syntax

```
Cell(ID.i, Text.s, Width.f, Hight.f, Border.i, Ln.i, Align.s, Fill.i, Link.i)
```

### Beschreibung

Platziert an der aktuellen Position (↩ linke obere Ecke) eine Zelle mit optionalen Rahmen, Hintergrundfarbe und Text auf der Seite. Der Text kann ausgerichtet oder zentriert werden.

Es ist zudem möglich einen Link auf den Text setzen.

**Width** [Default: `#PB_Default`]

Breite der Zelle festlegen. (`#PB_Default` = Seitenbreite)

**Height** [Default: `#PB_Default`]

Höhe der Zelle (Textzeile) festlegen. (`#PB_Default` = Zeichensatzhöhe)

**Border** [Default: `#False`]

Rahmen um die Zelle zeichnen.

`#True` / `#False`

`#LeftBorder` | `#TopBorder` | `#RightBorder` | `#BottomBorder`

**Ln** [Default: `#Right`]

Position für nächste Ausgabe festlegen:

`#Right` oder `0` ( rechts von der aktuellen Position fortsetzen )

`#NextLine` oder `1` ( Beginn der nächsten Zeile )

`#Below` oder `2` ( Unterhalb der aktuelle Position fortsetzen )

**Align** [Default: `#LeftAlign`]

Zentrierung oder Textausrichtung festlegen:

`#RightAlign` oder `"R"` ( linksbündig )

`#LeftAlign` oder `"L"` ( rechtsbündig )

`#ForcedJustified` oder `"F"` ( Blocksatz auch bei zu großen Wortabständen )

`#Justified` oder `"J"` ( Blocksatz )

`#CenterAlign` oder `"C"` ( zentriert )

**Fill** [Default: `#False`]

Farbiger anstatt transparenter Hintergrund: `#True` / `#False`

**Link** [Default: `#NoLink`]

Von Links, Annotations oder Actions zurückgegebene `LinkID`.



## PDF::Close

### Syntax

```
Save(ID.i, FileName.s)
```

### Beschreibung

Beendet die Dokumentenerstellung und erzeugt eine PDF-Datei mit dem entsprechenden Namen.

#### FileName

*Dateiname (und Pfad) für das erzeugte PDF-Dokument*

## PDF::DividingLine

### Syntax

```
DividingLine(ID.i, X.f, Y.f, Width.f)
```

### Beschreibung

Platziert an den angegebenen Koordinaten eine Trennlinie auf der Seite.

**X** [Default: #PB\_Default]

*Horizontale Position der Linie. (#PB\_Default = aktuelle Position)*

**Y** [Default: #PB\_Default]

*Vertikale Position der Linie. (#PB\_Default = aktuelle Position)*

**Width** [Default: #PB\_Default]

*Breite der Trennlinie (#PB\_Default = Seitenbreite)*

## PDF::Image

### Syntax

```
Image(ID.i, FileName.s, X.f, Y.f, Width.f, Height.f, Link.i)
```

### Beschreibung

Platziert an den angegebenen Koordinaten ein Bild auf der Seite.

#### FileName

*Dateiname (und Pfad) des entsprechenden Bildes*

**X** [Default: #PB\_Default]

*Horizontale Position der Linie. (#PB\_Default = aktuelle Position)*

**Y** [Default: #PB\_Default]

*Vertikale Position der Linie. (#PB\_Default = aktuelle Position)*

**Width** [Default: #PB\_Default]

*Breite des Bildes festlegen. (#PB\_Default = Originalgröße)*

**Height** [Default: #PB\_Default]

*Höhe des Bildes festlegen. (#PB\_Default = Originalgröße)*

**Link** [Default: #NoLink]

*Von Links, Annotations oder Actions zurückgegebene LinkID.*

## PDF::Ln

### Syntax

```
Ln(ID.i, Height.f=#PB_Default)
```

### Beschreibung

Führt einen Zeilenumbruch durch bzw. fügt eine Leerzeile mit der angegebenen Höhe ein. Dabei wird die aktuelle Position auf den Anfang der (nächsten) Zeile gesetzt.

**Height** [Default: #PB\_Default]

*Legt die Position (Y) für die nächste Zeile bzw. die Höhe der Leerzeile fest. (#PB\_Default = Zeilenhöhe)*

## PDF::MultiCell

### Syntax

```
MultiCell(ID.i, Text.s, Width.f, Height.f, Border.i, Align.s, Fill.i, Indent.i=0, maxLine.i)
```

### Beschreibung

Platziert an der aktuellen Position (↖ linke obere Ecke) eine Zelle mit mehreren Textzeilen auf der Seite.

#### Width

Breite der Zelle festlegen. (`#PB_Default` = Seitenbreite)

#### Height

Höhe der gesamten Zelle festlegen

**Border** [Default: `#False`]

Rahmen um die Zelle zeichnen.

`#True` / `#False`

`#LeftBorder` | `#TopBorder` | `#RightBorder` | `#BottomBorder`

**Align** [Default: `#LeftAlign`]

Zentrierung oder Textausrichtung festlegen:

`#RightAlign` oder `"R"` ( linksbündig )

`#LeftAlign` oder `"L"` ( rechtsbündig )

`#ForcedJustified` oder `"F"` ( Blocksatz auch bei zu großen Wortabständen )

`#Justified` oder `"J"` ( Blocksatz )

`#CenterAlign` oder `"C"` ( zentriert )

**Fill** [Default: `#False`]

Farbiger anstatt transparenter Hintergrund: `#True` / `#False`

**Indent** [Default: `0`]

Einfückung für die erste Zeile festlegen

**maxLine** [Default: `0`]

Maximale Zeilenzahl festlegen.

Wenn die maximale Zeilenzahl überschritten wird der restliche Text von der Prozedur zurückgegeben.

## PDF::MultiCellList

### Syntax

```
MultiCellList(ID.i, Text.s, Width.f, Height.f, Border.i, Align.s, Fill.i, Char.s)
```

### Beschreibung

Platziert an der aktuellen Position (↩ linke obere Ecke) eine Zelle mit einer Liste (mehreren Textzeilen) auf der Seite.

#### Width

Breite der Zelle festlegen. (`#PB_Default` = Seitenbreite)

#### Height

Höhe der gesamten Zelle festlegen

**Border** [Default: `#False`]

Rahmen um die Zelle zeichnen.

`#True` / `#False`

`#LeftBorder` | `#TopBorder` | `#RightBorder` | `#BottomBorder`

**Align** [Default: `#Justified`]

Zentrierung oder Textausrichtung festlegen:

`#RightAlign` oder `"R"` ( linksbündig )

`#LeftAlign` oder `"L"` ( rechtsbündig )

`#ForcedJustified` oder `"F"` ( Blocksatz auch bei zu großen Wortabständen )

`#Justified` oder `"J"` ( Blocksatz )

`#CenterAlign` oder `"C"` ( zentriert )

**Fill** [Default: `#False`]

Farbiger anstatt transparenter Hintergrund: `#True` / `#False`

**Char** [Default: `#Bullet$`]

Zeichen, das für den Spiegelstrich verwendet werden soll. (`#Bullet$` = `"•"`)

## PDF::PlaceText

### Syntax

```
PlaceText(ID.i, Text.s, X.f, Y.f)
```

### Beschreibung

Platziert den Text an den angegebenen Koordinaten auf der Seite.

**X** [Default: #PB\_Default]

*Horizontale Position des Textes. (#PB\_Default = aktuelle Position)*

**Y** [Default: #PB\_Default]

*Vertikale Position des Textes (#PB\_Default = aktuelle Position)*

## PDF::Rotate

### Syntax

```
Rotate(ID.i, Angle.f, X.f, Y.f)
```

### Beschreibung

Führt eine Drehung um den angegebenen Mittelpunkt durch. Die Rotation wirkt sich auf alle Elemente aus, die danach ausgegeben werden. Bei einer neuen Seite wird die Rotation auf aber wieder auf 0 gesetzt.

### Angle

*Drehwinkel in Grad festlegen.*

**X** [Default: #PB\_Default]

*Horizontale Position des Drehpunktes. (#PB\_Default = aktuelle Position)*

**Y** [Default: #PB\_Default]

*Vertikale Position des Drehpunktes (#PB\_Default = aktuelle Position)*

## PDF::SubWrite

### Syntax

```
SubWrite(ID.i, Text.s, Height.f, SubFontSize.i, SubOffset.f, Link.i)
```

### Beschreibung

Gibt wie **PDF::Write** den Text an der aktuellen Position aus. Dabei wird die Größe des Zeichensatzes verändert (z.B. für Initialen) und zusätzlich kann der Text mittels Offset hochgestellt (  $m^2$  ) bzw. tiefgestellt (  $H_2O$  ) werden.

**Height** [Default: **#PB\_Default**]

Zeilenhöhe festlegen. (**#PB\_Default** = Zeichensatzhöhe)

**SubFontSize** [Default: **12**]

Zeichensatzgröße in Point festlegen.

**SubOffset** [Default: **0**]

Offset in Points für den Text (positiv: hochgestellt / negativ: tiefgestellt)

**Link** [Default: **#NoLink**]

Von Links, Annotations oder Actions zurückgegebene **LinkID**.

## PDF::Write

### Syntax

```
Write(ID.i, Text.s, Height.f, Label.s, Link.i)
```

### Beschreibung

Erzeugt eine Zelle (rechteckiger Bereich) mit optionalen Rahmen, Hintergrundfarbe und Text.

**Height** [Default: **#PB\_Default**]

Zeilenhöhe festlegen. (**#PB\_Default** = Zeichensatzhöhe)

**Label**

Label der mit `AddGotoLabel()` definiert wurde. (anstatt **LinkID**)

**Link** [Default: **#NoLink**]

Von Links, Annotations oder Actions zurückgegebene **LinkID**.

## PDF::TruncateCell

### Syntax

`TruncateCell(ID.i, Text.s, Width.f, Hight.f, Border.i, Ln.i, Align.s, Fill.i, TruncText.s, Label.s, Link.i)`

### Beschreibung

Platziert an der aktuellen Position ( $\Leftarrow$  linke obere Ecke) eine Zelle mit optionalen Rahmen, Hintergrundfarbe und Text auf der Seite. Wenn der Text die angegebene Breite überschreitet, wird dieser abgeschnitten und der restliche Text wird von der Prozedur zurückgegeben. Der Text kann ausgerichtet oder zentriert werden. Es ist zudem möglich einen Link auf den Text setzen.

**Width** [Default: `#PB_Default`]

Breite der Zelle festlegen. (`#PB_Default` = Seitenbreite)

**Height** [Default: `#PB_Default`]

Höhe der Zelle (Textzeile) festlegen. (`#PB_Default` = Zeichensatzhöhe)

**Border** [Default: `#False`]

Rahmen um die Zelle zeichnen.

`#True` / `#False`

`#LeftBorder` | `#TopBorder` | `#RightBorder` | `#BottomBorder`

**Ln** [Default: `#Right`]

Position für nächste Ausgabe festlegen:

`#Right` (0): rechts von der aktuellen Position fortsetzen.

`#NextLine` (1): Beginn der nächsten Zeile.

`#Below` (2): Unterhalb der aktuelle Position fortsetzen.

**Align** [Default: `#LeftAlign`]

Zentrierung oder Textausrichtung festlegen:

`#RightAlign` oder `"R"` (linksbündig)

`#LeftAlign` oder `"L"` (rechtsbündig)

`#ForcedJustified` oder `"F"` (Blocksatz auch bei zu großen Wortabständen)

`#Justified` oder `"J"` (Blocksatz)

`#CenterAlign` oder `"C"` (zentriert)

**Fill** [Default: `#False`]

Farbiger anstatt transparenter Hintergrund: `#True` / `#False`

**TruncText** [Default: `"..."`]

Zeichenkette die am Ende des abgeschnittenen Textes angehängt wird.

### Label

Label der mit `AddGotoLabel()` definiert wurde. (anstatt `LinkID`)

**Link** [Default: `#NoLink`]

Von Links, Annotations oder Actions zurückgegebene `LinkID`.

## 3. Basis – Einstellungen

### 3.1 Parameter setzen

#### PDF::SetAutoPageBreak

##### Syntax

```
SetAutoPageBreak(ID.i, Flag.i, Margin.f)
```

##### Beschreibung

Aktivieren bzw. deaktivieren des automatischen Seitenumbruchs. Dieser ist standardmäßig aktiviert (Margin: 2cm).

##### Flag

Automatischer Zeilenumbruch: **#True** / **#False**

**Margin** [Default: 0]

Abstand zum Seitenende

#### PDF::SetColorRGB

##### Syntax

```
SetColorRGB(ID.i, ColorTyp.i, Red.f, Green.f=#PB_Default, Blue.f=#PB_Default)
```

##### Beschreibung

Farbe mittels RGB festlegen.

##### ColorTyp

**#TextColor** (Farbe für Textausgaben)

**#DrawColor** (Farbe für Zeichenausgaben)

**#FillColor** (Farbe für Hintergründe bzw. Füllfarbe bei Zeichenausgaben)

**Red** (0 - 255)

Rot - Anteil der Farbe bzw. Graustufen (falls **Green** = **#PB\_Default** und **Blue** = **#PB\_Default** ist).

**Green** (0 - 255)

Grün - Anteil der Farbe

**Blue** (0 - 255)

Blau - Anteil der Farbe



## PDF::SetDashedLine

### Syntax

```
SetDashedLine(ID.i, DashLength.i, DashSpacing.i)
```

### Beschreibung

Abstände und Strichlänge für gestrichelte Linien bzw.. Rechtecke festlegen.

#### DashLength

*Strichlängeder gestrichelten Line.*

#### DashSpacing

*Abstand zwischen den Strichen.*

## PDF::SetFont

### Syntax

```
SetFont(ID.i, Family.s, Style.s, Size.i=0)
```

### Beschreibung

Den Zeichensatz für die nachfolgenden Textausgaben festlegen.

**Family** [Default: `""`]

*Standard- oder eingebetteten Zeichensatz mit dem Schriftnamen auswählen ( `""` = aktueller Zeichensatz )*

**Courier**

**Helvetica**    oder **Arial**

**Times-Roman** oder **Times**

**Symbol**

**ZapfDingbats**

**Style** [Default: `""`]

*Die Attributte könne auch kombiniert werden (z.B. **BI** = fett & kursiv).*

**B**    ( fett )

**I**    ( kursiv )

**U**    ( unterstrichen )

**Size** [Default: `#PB_Default`]

*Schriftgröße in Points ( `#PB_Default` = aktuelle Schriftgröße )*

## PDF::SetFontSize

### Syntax

```
SetFontSize(ID.i, Size.i)
```

### Beschreibung

Die Zeichensatzgröße (Points) für die nachfolgenden Textausgaben ändern.

#### Size

*Schriftgröße in Points*

## PDF::SetPageNumbering

### Syntax

```
SetPageNumbering(ID.i, Flag.i)
```

### Beschreibung

Automatische Seitennummerierung aktivieren: **#True** / **#False**

## PDF::SetLineThickness

### Syntax

```
SetLineThickness(ID.i, Width.f)
```

### Beschreibung

Strichdicke für Linien festlegen. ( Standard: 0,2mm )

## PDF::SetLocales

### Syntax

```
SetLocales(ID.i, Type.i, Value.s)
```

### Beschreibung

#### Type

<b>#Language</b>	(Standard: " <b>DE</b> ")
<b>#TimeZoneOffset</b>	(Standard: " <b>+1</b> ")
<b>#DecimalPoint</b>	(Standard: " <b>,</b> ")
<b>#ThousandSeparator</b>	(Standard: " <b>.</b> ")

## PDF::SetMargin

### Syntax

```
SetMargin(ID.i, Type.i, Value.i)
```

### Beschreibung

#### Type

#TopMargin (Oberer Rand)  
#LeftMargin (Linker Rand)  
#RightMargin (Rechter Rand)  
#CellMargin (Abstand zwischen Text und Zellenrand)

## PDF::SetPageMargins

### Syntax

```
SetPageMargins(ID.i, LeftMargin.f, TopMargin.f, RightMargin.f)
```

### Beschreibung

Breite des linken, oberen und rechten Randes der Seite festlegen.

**RightMargin** [Default: #PB\_Default]

*#PB\_Default = Breite des linken Randes*

## PDF::SetPosXY

### Syntax

```
SetPosXY(ID.i, X.f, Y.f)
```

### Beschreibung

Horizontale (X) und vertikale (Y) Position für nachfolgende Ausgaben festlegen.

## PDF::SetPosX

### Syntax

```
SetPosX(ID.i, X.f)
```

### Beschreibung

Horizontale (X) Position für nachfolgende Ausgaben festlegen.

## PDF::SetPosY

### Syntax

```
SetPosY(ID.i, Y.f, ResetX.i)
```

### Beschreibung

Vertikale (Y) Position für nachfolgende Ausgaben festlegen.

## PDF::SetInfo

### Syntax

```
SetInfo(ID.i, Type.i, Value.s)
```

### Beschreibung

Metadaten mit Informationen über das Dokument festlegen.

### Type

#Author	( Autor )
#Titel	( Titel des Dokumentes )
#Subject	( Thema )
#Keywords	( Stichwörter getrennt durch Leerzeichen )
#Creator	( Ersteller des Dokummentes )

## PDF::SetWordSpacing

### Syntax

```
SetWordSpacing(ID.i, WordSpacing.f)
```

### Beschreibung

Den Wortabstand verändern (Standard: Breite eines Leerzeichens).

## 3.2 Parameter auslesen

## PDF::GetFontSize

### Syntax

```
GetFontSize(ID.i, Type.i)
```

### Beschreibung

**Type** [Default: **#Point**]

**#Point** (Schriftgröße in Points)

**#Unit** (Schriftgröße in der gewählten Einheit)

## PDF::GetMargin

### Syntax

```
GetMargin(ID.i, Type.i)
```

### Beschreibung

#### Type

**#TopMargin** (Oberer Rand)

**#LeftMargin** (Linker Rand)

**#RightMargin** (Rechter Rand)

**#CellMargin** (Abstand zwischen Text und Zellenrand)

## PDF::GetMultiCellLines

### Syntax

```
GetMultiCellLines(ID.i)
```

### Beschreibung

Aktuelle Anzahl der Zeilen in **PDF::MultiCell** bzw. **PDF::MultiCellList**.

## PDF::GetPageHeigth

### Syntax

```
GetPageHeigth(ID.i)
```

### Beschreibung

Höhe der aktuellen Seite

## PDF::GetPageNumber

### Syntax

```
GetPageNumber(ID.i, TOC.i)
```

### Beschreibung

Seitennummer der aktuellen Seite.

**TOC** [Default: **#False**]

Seitennummer des Inhaltsverzeichnisses (TOC) anstatt der Seite.

## PDF::GetPageWidth

### Syntax

```
GetPageWidth(ID.i)
```

### Beschreibung

Breite der aktuellen Seite.

## PDF::GetPosX

### Syntax

```
GetPosX(ID.i)
```

### Beschreibung

Aktuelle horizontale Position (X) auf der Seite.

## PDF::GetPosY

### Syntax

```
GetPosY(ID.i)
```

### Beschreibung

Aktuelle vertikale Position (Y) auf der Seite.

## PDF::GetScaleFactor

### Syntax

```
GetScaleFactor(ID.i)
```

### Beschreibung

Der Skalierungsfaktor für die im Dokument genutzte Einheit. ( z.B. mm = Points x ScaleFactor )

## PDF::GetStringWidth

### Syntax

```
GetStringWidth(ID.i, String.s)
```

### Beschreibung

Die Breite des Strings/Textes in der vom Dokument verwendeten Einheit. (Standard: mm)

## PDF::GetWordSpacing

### Syntax

```
GetWordSpacing(ID.i)
```

### Beschreibung

Breite der Wortabstände.



## 4. Fortgeschrittene Befehle

### PDF::AddEntryTOC

#### Syntax

```
AddEntryTOC(ID.i, Text.s, Level.i)
```

#### Beschreibung

Eintrag zum Inhaltsverzeichnis hinzufügen.

#### Text

*Text der im Inhaltsverzeichnis erscheinen soll.*

**Level** [Default: 0]

*Ebene im Inhaltsverzeichnis (oberste Ebene = 0)*

### PDF::AddGotoAction

#### Syntax

```
AddGotoAction(ID.i, Page.i=#PB_Default, PosY.i=#PB_Default)
```

#### Beschreibung

Springt zu der entsprechenden (bereits existierenden!) Seite. Gibt die dafür nötige **LinkID** zurück.

#### Title

*Titel für die Anmerkung.*

**Page** [Default: #PB\_Default]

*Seite zu der gesprungen werden soll. (#PB\_Default = aktuelle Position)*

**Y** [Default: #PB\_Default]

*Vertikale Position auf der Zielseite. (#PB\_Default = aktuelle Position)*

## PDF::AddFileLink

### Syntax

```
AddFileLink(ID.i, EmbedID.i, Title.s, Description.s, Icon.s, IconW.f, IconH.f, X.f)
```

### Beschreibung

Einen Link zu einer eingebetteten Datei hinzufügen. Gibt die dafür nötige LinkID zurück.

#### EmbedID

Die *EmbedID*, die von *PDF::AddFile* zurückgegeben wird.

**Title** [Default: ""]

Titel für den Link (z.B. Dateiname).

**Description** [Default: ""]

Beschreibung der eingebetteten Datei, auf die der Link verweist.

**Icon** [Default: #PushPinIcon]

Icon, das für die Anzeige des Links verwendet werden soll.

#GraphIcon        oder   **Graph**  
#PaperClipIcon   oder   **Paperclip**  
#PushPinIcon     oder   **PushPin**  
#TagIcon          oder   **Tag**

**IconW** [Default: 10]

Breite des Icons

**IconH** [Default: 10]

Höhe des Icons

**X** [Default: #PB\_Default]

Horizontale Position der Zielseite. (#PB\_Default = aktuelle Position)

## PDF::AddGotoLabel

### Syntax

```
AddGotoAction(ID.i, Label.s)
```

### Beschreibung

Definiert an der aktuellen Positon einen Label, zu dem gesprungen werden kann. Gegenüber eine LinkID muss ein Label nicht bereits vor dem Aufruf definiert werden, sondern kann auch auf einer Position danach befinden.

## PDF::AddLaunchAction

### Syntax

```
AddLaunchAction(ID.i, FileName.s, Y.f, Action.s)
```

### Beschreibung

Ruft eine externe Datei auf, wenn das Öffne des Datei-Typs vom Betriebssystem bzw. dem PDF-Betrachter unterstützt wird. Gibt die dafür nötige **LinkID** zurück.

### FileName

*Dateiname (und Pfad) der externen Datei. Der Dateityp muss im Betriebssystem mit einem passende Programm verknüpft sein, damit dieser geöffnet werden kann.*

**Y** [Default: **#PB\_Default**]

*Vertikale Position der Anmerkung (Aktion) auf der Seite. (**#PB\_Default** = aktuelle Position)*

**Action** [Default: **#OpenAction**]

**#OpenAction** oder **open**

**#PrintAction** oder **print**

## PDF::AddLinkURL

### Syntax

```
AddLinkURL(ID.i, URL.s)
```

### Beschreibung

Link zu einer Internetadresse auf der Seite hinzufügen. Gibt die dafür nötige LinkID zurück.

### URL

*Adresse der Internetseite festlegen. Es wird die komplette URL einschliesslich Protokoll ("http://" etc.) benötigt.*

## PDF::AddTextNote

### Syntax

```
AddTextNote(ID.i, Title.s, Content.s, Icon.s, Y.f)
```

### Beschreibung

Fügt eine Text - Anmerkung (vgl. Tooltip) zur Seite hinzu. Gibt die dafür nötige LinkID zurück.

#### Title

*Titel für die Anmerkung.*

#### Content

*Inhalt der Anmerkung.*

**Y** [Default: `#PB_Default`]

*Vertikale Position der Anmerkung auf der Seite. (`#PB_Default` = aktuelle Position)*

**Icon** [Default: `#PB_NoteIcon`]

*Icon, das für die Anzeige des der Anmerkung verwendet werden soll.*

<code>#CommentIcon</code>	oder	<code>Comment</code>
<code>#KeyIcon</code>	oder	<code>Key</code>
<code>#NoteIcon</code>	oder	<code>Note</code>
<code>#HelpIcon</code>	oder	<code>Help</code>
<code>#NewParagraphIcon</code>	oder	<code>NewParagraph</code>
<code>#ParagraphIcon</code>	oder	<code>Paragraph</code>
<code>#Insert</code>	oder	<code>Insert</code>

## PDF::EmbedFile

### Syntax

```
EmbedFile(ID.i, Filename.s, Description.s)
```

### Beschreibung

Ein Datei im PDF-Dokument einbetten.

#### Filename

*Dateiname (und Pfad) der entsprechenden Datei.*

**Description** [Default: `""`]

*Beschreibung zur Datei.*

## PDF::EmbedFont

### Syntax

```
EmbedFont(ID.i, Filename.s, Family.s, Style.s, Flags.i)
```

### Beschreibung

Einen Zeichensatz im PDF-Dokument einbetten und zur Verfügung stellen (⇒ **PDF::SetFont** ).  
Die Prozedur gibt den internen Namen aus dem TTF-Header zurück.

#### Filename

*Dateiname (und Pfad) des TrueType-Fonts.*

**Family** [Default: **""**]

*Name der für den Aufruf mit **PDF::SetFont** verwendet werden soll. ( "" = interner Name des TTF-Headers)*

**Style** [Default: **""**]

*Style der für den Aufruf mittels **PDF::SetFont** verwendet werden soll. ( "" für arial.ttf / "B" für arial**bd**.ttf / "I" für arial**i**.ttf ).*

**Flags** [Default: **#False**]

**#Unicode** (Unicode – Zeichensatz mit > 256 und ≤ 65535 Zeichen )

## PDF::EmbedJavaScript

### Syntax

```
EmbedJavaScript(ID.i, Script.s, Name.s)
```

### Beschreibung

Einen String mit JavaScript-Code in das Dokument einbetten.

## PDF::EmbedJavaScriptFile

### Syntax

```
EmbedJavaScriptFile(ID.i, FileName.s, Name.s)
```

### Beschreibung

Eine Datei mit JavaScript-Code in das Dokument einbetten.

## PDF::EnableTOCNums

### Syntax

```
EnableTOCNums(ID.i, Flag.i=#True)
```

### Beschreibung

Aktivieren bzw. Deaktivieren der Seitennummerierung für das Inhaltsverzeichnis.

## PDF::EnableFooter

### Syntax

```
EnableFooter(ID.i, Flag.i=#True)
```

### Beschreibung

Aktivieren bzw. Deaktivieren der Procedure für die Fußzeile..

## PDF::EnableHeader

### Syntax

```
EnableHeader(ID.i, Flag.i=#True)
```

### Beschreibung

Aktivieren bzw. Deaktivieren der Procedure für die Kopfzeile.

## PDF::EscapeText

### Syntax

```
EscapeText(ID.i, Text.s)
```

### Beschreibung

Gibt eine für PDF maskierte Version des Textes zurück. ( z.B. **#LF\$** ⇒ **\n** )

## PDF::InsertTOC

### Syntax

```
InsertTOC(ID.i, Page.i=1, Label.s="", LabelFontSize.i=20, EntryFontSize.i=10, FontFamily.s="Times")
```

### Beschreibung

Ein Inhaltsverzeichnis in das Dokument einfügen.

**Page** [Default: **1**]

Seite auf der das Inhaltsverzeichnis eingefügt werden soll.

**Label** [Default: **""**]

Überschrift für das Inhaltsverzeichnis ( **""** ⇒ Überschrift gemäß **#DefaultLanguage** )

**LabelFontSize** [Default: **20**]

Schriftgröße für die Überschrift

**EntryFontSize** [Default: **10**]

Schriftgröße für die Einträge

**FontFamily** [Default: **Times**]

Schrift, die für das Inhaltsverzeichnis verwendet werden soll

## PDF::ImageMemory

### Syntax

```
ImageMemory(ID.i, Name.s, *Memory, MemSize.i, Format.i, X.f, Y.f, Width.f, Height.f, Link.i)
```

### Beschreibung

Platziert an den angegebenen Koordinaten ein im Speicher abgelegtes Bild auf der Seite.

**FileName**

Dateiname (und Pfad) des entsprechenden Bildes

**Name**

Dateiname für das Bild im Speicher

**\*Memory**

Pointer auf die Speicheradresse des Bildes

**MemSize**

Größe der Bilddaten im Speicher (Bytes)

**Format**

**#Image\_PNG**

**#Image\_JPEG**

**#Image\_JPEG2000**

**X** [Default: **#PB\_Default**]

Horizontale Position der Linie. (**#PB\_Default** = aktuelle Position)

**Y** [Default: **#PB\_Default**]

Vertikale Position der Linie. (**#PB\_Default** = aktuelle Position)

**Width** [Default: **#PB\_Default**]

Breite des Bildes festlegen. (**#PB\_Default** = Originalgröße)

**Height** [Default: **#PB\_Default**]

Höhe des Bildes festlegen. (**#PB\_Default** = Originalgröße)

**Link** [Default: **#False**]

Von Links, Annotations oder Actions zurückgegebene **LinkID**.

## PDF::LinkArea

### Syntax

**LinkArea**(ID.i, X.f, Y.f, Width.f, Height.f, Label.s, Link.i)

### Beschreibung

Definiert einen rechteckigen Bereich auf der Seite als Link. Kann für Ausgaben mittels **PDF::Image** oder **PDF::Write** verwendet werden.

**X**

Horizontale Position der Linie.

**Y**

Vertikale Position der Linie.

**Width**

Breite des Bildes festlegen.

**Heigh**

Höhe des Bildes festlegen

**Label**

Label der mit **AddGotoLabel()** definiert wurde. (anstatt **LinkID**)

**Link** [Default: **#NoLink**]

Von Annotations oder Actions zurückgegebene **LinkID**.



## 5. Fortgeschrittene Einstellungen

### PDF::GetErrorCode

#### Syntax

```
GetErrorCode(ID.i)
```

#### Beschreibung

Liefert den internen Errorcode zurück.

### PDF::GetErrorMessage

#### Syntax

```
GetErrorMessage(ID.i)
```

#### Beschreibung

Liefert die letzte Fehlermeldung zurück.

### PDF::GetNumbering

#### Syntax

```
GetNumbering(ID.i, TOC.i)
```

#### Beschreibung

Liefert zurück, ob die Seitennummerierung bzw. die Seitennummern für das Inhaltsverzeichnis (**#True**) aktiviert sind.

### PDF::GetObjectNum

#### Syntax

```
GetObjectNum(ID.i)
```

#### Beschreibung

Liefert die Nummer des aktuellen PDF-Objektes (z.B. Objektnummer der aktuellen Seite) zurück.

## PDF::

### Syntax

```
SetAliasTotalPages(ID.i, Alias.s)
```

### Beschreibung

Definiert einen Alias für die Gesamtzahl der Seiten der dann im benutzerdefinierten Footer verwendet werden kann. ( z.B. "Seite 2 / 5" )

## PDF::

### Syntax

```
SetColorCMYK(ID.i, ColorTyp.i, Cyan.f, Magenta.f, Yellow.f, Black.f)
```

### Beschreibung

Farbe mittels CMYK festlegen.

### ColorTyp

**#TextColor** ( Farbe für Textausgaben )

**#DrawColor** ( Farbe für Zeichenausgaben )

**#FillColor** ( Farbe für Hintergründe bzw. Füllfarbe bei Zeichenausgaben )

**Cyan** (0.0 - 1.0)

Blaugrün - Anteil der Farbe

**Magenta** (0.0 - 1.0)

Purpurrot - Anteil der Farbe

**Yellow** (0.0 - 1.0)

Gelb - Anteil der Farbe

**Black** (0.0 - 1.0)

Schwarz - Anteil der Farbe

## PDF::SetEncryption

### Syntax

```
SetEncryption(ID.i, User.s, Owner.s, Permission.q)
```

### Beschreibung

Dokument verschlüsseln und Berechtigungen festlegen.

#### User

*Passwort zum Öffnen des Dokumentes mit den festgelegten Berechtigungen / Einschränkungen.*

#### Owner

*Passwort des Erstellers des Dokumentes.*

**Permission** [Default: **#False**]

**#PrintAccess** ( Drucken des Dokumentes aber nicht unbedingt in höchster Qualität )

**#ModifyAccess** ( Aktualisieren des Dokumenteninhaltes )

**#CopyAccess** ( Kopieren von Text oder Grafiken )

**#AddAccess** ( Hinzufügen oder ändern von Anmerkungen bzw. von Formularfeldern )

**#FormAccess** ( Ausfüllen bestehender Formularfelder )

**#Extract** ( Extrahieren von Text und Grafiken )

**#Assemble** ( Einfügen, drehen oder Löschen von Seiten / hinzufügen von Lesezeichen )

**#DigitalCopy** ( Drucken als originalgetreue digitale Kopie )

## PDF::SetPageCompression

### Syntax

```
SetPageCompression(ID.i, Flag.i)
```

### Beschreibung

Aktivieren bzw. deaktivieren der Kompression für Seiteninhalte: **#True** / **#False**

## PDF::SetHeaderProcedure

### Syntax

```
SetHeaderProcedure(ID.i, *ProcAddress, *StructAddress=NULL)
```

### Beschreibung

Übergibt den Pointer auf eine benutzerdefinierte Header-Prozedur (Kopfzeile) und ggf. den Pointer auf eine Struktur.

## PDF::SetOpenAction

### Syntax

```
Set(ID.i, Zoom.s, Page.i)
```

### Beschreibung

Legt fest, was beim Öffnen des Dokumentes geschehen soll .

#### Zoom

*Zoom in Prozent (z.B. "50%") oder eine der folgenden Konstanten:*

`#FullPage`

`#PageWidth`

#### Page

*Gehe zu dieser Seite (z.B. Überspringen von Titel und Inhaltsverzeichnis usw.)*

## PDF::SetFooterProcedure

### Syntax

```
SetFooterProcedure(ID.i, *ProcAddress, *StructAddress=NULL)
```

### Beschreibung

Übergibt den Pointer auf eine benutzerdefinierte Footer-Prozedur (Fußzeile) und ggf. den Pointer auf eine Struktur.

## PDF::SetPageLayout

### Syntax

```
SetPageLayout(ID.i, Mode.s)
```

### Beschreibung

Legt fest, wie das Dokument im PDF-Betrachter angezeigt werden soll.

**Mode** [Default: #SinglePage]

#SinglePage	( einzelne Seiten )
#Continuous	( Seiten fortlaufend )
#TwoColumn	( zwei Spalten - ungerade Seite links )
#TwoColumnRight	( zwei Spalten - ungerade Seite rechts )
#TwoPages	( zwei Seiten aufeinmal - ungerade Seite links )
#TwoPagesRight	( zwei Seiten aufeinmal - ungerade Seite rechts )

## PDF::SetPageMode

### Syntax

```
SetPageLayout(ID.i, Mode.s)
```

### Beschreibung

Legt fest, wie das Dokument im PDF-Betrachter angezeigt werden soll.

**Mode** [Default: #SinglePage]

#None	( weder Lesezeichen noch Miniaturansichten )
#Outlines	( Lesezeichen anzeigen )
#Thumbs	( Miniaturansichten anzeigen )
#FullScreen	( Fullscreen-Modus )
#OptionalContent	( Optionalen Inhalt anzeigen )
#Attachmentst	( Anlagen anzeigen )

## PDF::SetViewerPreferences

### Syntax

```
SetViewerPreferences(ID.i, Flags.i)
```

### Beschreibung

Voreinstellungen für den PDF-Betrachter.

### Preferences

#HideMenubar	( <i>Menüleiste verbergen</i> )
#HideToolbar	( <i>Werkzeugleiste verbergen</i> )
#HideWinUI	( <i>Fensterelemente verbergen</i> )
#DisplayTitle	( <i>Dokumententitel anstatt Dateinamen anzeigen</i> )
#CenterWindow	( <i>Fenster auf dem Bildschirm zentrieren</i> )
#FitWindow	( <i>Fenstergröße an Bildschirm anpassen</i> )

## 6. Befehle für die Formularerstellung

### PDF::ButtonField

#### Syntax

```
ButtonField(ID.i, Titel.s, State.i, Flags.i, Width.f, Height.f, Ln.i)
```

#### Beschreibung

Einfügen einer Checkbox oder eines Radio-Buttons.

#### Titel

*Titel bzw. Bezeichnung für dieses Feld.*

**Status** [Default: `#False`]

Anfangsstatus der Checkbox bzw. des Radiobuttons (aktiviert/deaktiviert).

**Flags** [Default: `#False`]

`#Form_CheckBox`

`#Form_RadioButton`

`#Form_NoToggleToOff` (immer nur ein Radiobutton kann ausgewählt werden )

**Width** [Default: `#PB_Default`]

Breite der Zelle festlegen. (`#PB_Default` = Seitenbreite)

**Height** [Default: `#PB_Default`]

Höhe der Zelle (Textzeile) festlegen. (`#PB_Default` = Zeichensatzhöhe)

**Ln** [Default: `#Right`]

Position für nächste Ausgabe festlegen:

`#Right` oder `0` (rechts von der aktuellen Position fortsetzen )

`#NextLine` oder `1` (Beginn der nächsten Zeile )

`#Below` oder `2` (Unterhalb der aktuelle Position fortsetzen )

## PDF::ChoiceField

### Syntax

```
ChoiceField(ID.i, Titel.s, Value.s, Options.s, Index.i, Flags.i, Width.f, Height.f, Ln.i)
```

### Beschreibung

Einfügen eines Auswahlfeldes (ComboBox).

#### Titel

*Titel bzw. Bezeichnung für dieses Feld.*

#### Value

*Standardmäßig vorgebener Wert für das Feld*

#### Options

*Auswahloptionen vorgeben. Die einzelnen Werte müssen durch ein Leerzeichen voneinander getrennt sein.*

**Index** [Default: `#PB_Default`]

*Standardmäßig ausgewähltes Feld festlegen.*

**Flags** [Default: `#False`]

`#Form_ComboBox`

`#Form_Edit` ( *ComboBox ist editierbar* )

`#Form_CommitOnSelChange` ( *sofortige Auswahl durch Mausklick* )

**Width** [Default: `#PB_Default`]

*Breite der Zelle festlegen. (#PB\_Default = Seitenbreite)*

**Height** [Default: `#PB_Default`]

*Höhe der Zelle (Textzeile) festlegen. (#PB\_Default = Zeichensatzhöhe)*

**Ln** [Default: `#NextLine`]

*Position für nächste Ausgabe festlegen:*

`#Right` oder `0` ( *rechts von der aktuellen Position fortsetzen* )

`#NextLine` oder `1` ( *Beginn der nächsten Zeile* )

`#Below` oder `2` ( *Unterhalb der aktuelle Position fortsetzen* )



## PDF::TextField

### Syntax

`TextField(ID.i, Titel.s, Text.s, Flags.i, Width.f, Height.f, Border.i, Ln.it, Align.s, Fill.i)`

### Beschreibung

Einfügen eines Textfeldes.

#### Titel

*Titel bzw. Bezeichnung für dieses Feld.*

#### Text

*Standardmäßig vorgegebener Text.*

**Flags** [Default: `#False`]

`#Form_MultiLine` ( Mehrzeiliges Textfeld )  
`#Form_DoNotSpellCheck` ( Bei Rechtschreibprüfung ignorieren )  
`#Form_DoNotScroll` ( Text darf nicht länger als das Textfeld sein )

**Width** [Default: `#PB_Default`]

*Breite der Zelle festlegen. (#PB\_Default = Seitenbreite)*

**Height** [Default: `#PB_Default`]

*Höhe der Zelle (Textzeile) festlegen. (#PB\_Default = Zeichensatzhöhe)*

**Border** [Default: `#False`]

*Rahmen um das Textfeld zeichnen.*

**Ln** [Default: `#Right`]

*Position für nächste Ausgabe festlegen:*

`#Right` oder `0` ( rechts von der aktuellen Position fortsetzen )  
`#NextLine` oder `1` ( Beginn der nächsten Zeile )  
`#Below` oder `2` ( Unterhalb der aktuelle Position fortsetzen )

**Align** [Default: `#LeftAlign`]

*Zentrierung oder Textausrichtung festlegen:*

`#RightAlign` oder `"R"` ( linksbündig )  
`#LeftAlign` oder `"L"` ( rechtsbündig )  
`#CenterAlign` oder `"C"` ( zentriert )

**Fill** [Default: `#False`]

*Farbiger anstatt transparenter Hintergrund: `#True` / `#False`*

## 7. Befehle zum Erzeugen geometrischer Formen

### PDF::DrawCircle

#### Syntax

```
DrawCircle(ID.i, X.f, Y.f, Radius.f, Style.s)
```

#### Beschreibung

Zeichnet an den angegebenen Koordinaten einen Kreis.

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly      oder "F"

#DrawAndFill   oder "DF"

### PDF::DrawEllipse

#### Syntax

```
DrawEllipse(ID.i, X.f, Y.f, xRadius.f, yRadius.f, Style.s)
```

#### Beschreibung

Zeichnet an den angegebenen Koordinaten eine Ellipse.

#### xRadius

*Horizontaler Radius der Ellipse.*

#### yRadius

*Vertikaler Radius der Ellipse.*

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly      oder "F"

#DrawAndFill   oder "DF"

## PDF::DrawGrid

### Syntax

```
DrawGrid(ID.i, Spacing.i)
```

### Beschreibung

Zeichnet auf der Seite ein hellblaues Gitter (z.B. zu Testzwecken).

**Spacing** [Default: #PB\_Default]

Abstand der Gitterlinien.

## PDF::DrawLine

### Syntax

```
DrawLine(ID.i, X1.f, Y1.f, X2.f, Y2.f)
```

### Beschreibung

Zeichnet an den angegebenen Koordinaten eine Linie.

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly       oder "F"

#DrawAndFill   oder "DF"

## PDF::DrawRectangle

### Syntax

```
DrawRectangle(ID.i, X.f, Y.f, Width.f, Height.f, Style.s)
```

### Beschreibung

Zeichnet an den angegebenen Koordinaten ein Rechteck.

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly       oder "F"

#DrawAndFill   oder "DF"

## PDF::DrawRoundedRectangle

### Syntax

```
DrawRoundedRectangle(ID.i, X.f, Y.f, Width.f, Height.f, Radius.f, Style.s="")
```

### Beschreibung

Zeichnet an den angegebenen Koordinaten eine Rechteck mit abgerundeten Ecken.

### Radius

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly      oder "F"

#DrawAndFill   oder "DF"

## PDF::DrawSector

### Syntax

```
DrawSector(ID.i, X.f, Y.f, Radius.f, startAngle.f, endAngle.f, Style.s, Clockwise.i, originAngle.f)
```

### Beschreibung

Zeichnet an den angegebenen Koordinaten ein Kreissegment (z.B. für Kreisdiagramme).

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly      oder "F"

#DrawAndFill   oder "DF"

**Clockwise** [Default: #True]

*im Uhrzeigersinn (#True) bzw. gegen den Uhrzeigersinn (#False)*

**originAngle** [Default: 90]

*Ursprungswinkel in Grad auf den sich die Angaben für den Start- und Endwinkel beziehen.*

## PDF::DrawTriangle

### Syntax

```
DrawTriangle(ID.i, X1.f, Y1.f, X2.f, Y2.f, X3.f, Y3.f, Style.s)
```

### Beschreibung

Zeichnet an den angegebenen Koordinaten eine Dreieck.

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly      oder "F"

#DrawAndFill   oder "DF"

## PDF::PathBegin

### Syntax

```
PathBegin(ID.i, X.f, Y.f)
```

### Beschreibung

Beginnt einen neuen Pfad (Zeichnung) an den angegebenen Koordinaten.

## PDF::PathEnd

### Syntax

```
PathEnd(ID.i, Style.s="")
```

### Beschreibung

Beendet den aktuellen Pfad und zeichnet gemäß den Pfadangaben eine Linie vom Startpunkt zum Endpunkt.

**Style** [Default: #PB\_DrawOnly]

#DrawOnly      oder "D"

#FillOnly      oder "F"

#DrawAndFill   oder "DF"

## PDF::PathArc

### Syntax

```
PathArc(ID.i, X1.f, Y1.f, X2.f, Y2.f, X3.f, Y3.f)
```

### Beschreibung

Fügt an den aktuellen Pfad-Koordinaten eine kubische Bezierkurve in den aktuellen Pfad ein, wobei (X3, Y3) den Endpunkt und (X1, Y1) und (X2, Y2) als Bezier-Kontrollpunkte verwendet werden.

## PDF::PathLine

### Syntax

```
PathLine(ID.i, X.f, Y.f)
```

### Beschreibung

Fügt an den aktuellen Pfad-Koordinaten eine Linie mit dem Endpunkt (X, Y) ein.

## 8. Befehle für Transformationen

### PDF::StartTransform

#### Syntax

```
StartTransform(ID.i)
```

#### Beschreibung

Startet eine oder mehrere Transformationen und muss somit vor der Verwendung von **PDF::Scale**, **PDF::Skew**, **PDF::Mirror**, **PDF::Translate** aufgerufen werden.

### PDF::StopTransform

#### Syntax

```
StopTransform(ID.i)
```

#### Beschreibung

Beendet die Transformation(en).

### PDF::MirrorHorizontal

#### Syntax

```
MirrorHorizontal(ID.i, X.f)
```

#### Beschreibung [X: #PB\_Default]

Horizontale Spiegelung, wobei X die Position der Spiegelachse bestimmt. (#PB\_Default = aktuelle Position)

### PDF::MirrorVertical

#### Syntax

```
MirrorVertical(ID.i, Y.f)
```

#### Beschreibung [Y: #PB\_Default]

Vertikale Spiegelung, wobei Y die Position der Spiegelachse bestimmt. (#PB\_Default = aktuelle Position)

## PDF::Scale

### Syntax

```
Scale(ID.i, Factor.f, X.f, Y.f)
```

**Beschreibung** [X / Y: #PB\_Default]

Vergrößerung um **Factor** (Default: **100**).

## PDF::ScaleHorizontal

### Syntax

```
ScaleHorizontal(ID.i, Factor.f, X.f, Y.f)
```

**Beschreibung** [X / Y: #PB\_Default]

Horizontale Vergrößerung um **Factor** (Default: **100**).

## PDF::ScaleVertical

### Syntax

```
ScaleVertical(ID.i, Factor.f, X.f, Y.f)
```

**Beschreibung**

Vertikale Vergrößerung um **Factor** (Default: **100**).

## PDF::SkewHorizontal

### Syntax

```
SkewHorizontal(ID.i, Angle.i, X.f, Y.f)
```

**Beschreibung** [X / Y: #PB\_Default]

Horizontale Verzerrung um den gewählten Winkel (**Angle**).



## PDF::SkewVertical

**Syntax** [X / Y: #PB\_Default]

```
SkewVertical(ID.i, Angle.i, X.f, Y.f)
```

### Beschreibung

Vertikale Verzerrung um den gewählten Winkel (**Angle**).

## PDF::Translate

**Syntax**

```
Translate(ID.i, moveX.f, moveY.f)
```

### Beschreibung

Verschieben in der Horizontalen (**moveX**) und der Vertikalen (**moveY**).

## PDF::TranslateHorizontal

**Syntax**

```
TranslateHorizontal(ID.i, moveX.f)
```

### Beschreibung

Verschieben in der Horizontalen (**moveX**).

## PDF::TranslateVertical

**Syntax**

```
TranslateVertical(ID.i, moveY.f)
```

### Beschreibung

Verschieben in der Vertikalen (**moveY**).

Embe:Embe

# Literaturverzeichnis

- Adobe Systems Incorporated, "**PDF Reference – Adobe Portable Document Format**" (Version 1.6), 2004
- INFORMATION TECHNOLOGY, "**JPEG 2000 IMAGE CODING SYSTEM**", 16. March 2000
- Guillaume Endignoux, "**How secure is PDF Encrypting**", 2. November 2016
- Yauhen Yakimenka, "**PDF Encryption – Research seminar in cryptography**", 15. December 2015
- John Whittington, "**PDF Explained**" (second release), 2012
- Leonard Rosenthol, "**Developing with PDF**" (First Release), 2014
- Thomas Merz, Olaf Drümmer, "**Die PostScript & PDF-Bibel**" (2. Auflage), 2002
- Christof Bürgi, "**Das PDF Howto**" ( <http://www.p2501.ch/pdf-howto> ), 26. March 2013
- Lucky Luke, "**PurePDF Purebasic UserLibrary**", Version 2.08