

Polybios 1.3

Making History In PDF

Andreas Falkenhahn

Inhaltsverzeichnis

1	Allgemeine Informationen	1
1.1	Einführung	1
1.2	Lizenzvereinbarung	1
1.3	Voraussetzung	2
1.4	Installation	2
2	Über Polybios	5
2.1	Danksagungen	5
2.2	Häufig gestellte Fragen	5
2.3	Zukunft	6
2.4	Geschichte	6
3	Anzeigen von PDFs	7
3.1	Übersicht	7
3.2	Laden von Seiten als Vektorpinsel	7
3.3	Laden von Seiten als Animation	8
4	Erstellen von PDFs	9
4.1	Koordinatensystem	9
4.2	Grafikmodus	9
4.3	Pfade zeichnen	10
4.4	Text ausgeben	11
4.5	Farben	12
4.6	Schriftarten	12
4.7	Base14-Schriftarten	13
4.8	Type1-Schriftarten	13
4.9	TrueType-Schriftarten	14
4.10	CID-Schriftarten	14
4.11	Kodierungen	15
5	Tutorial	19
5.1	Tutorial	19
6	Allgemeine Befehle	21
6.1	pdf.CloseDocument	21
6.2	pdf.CreateDocument	21
6.3	pdf.DeviceToPage	22
6.4	pdf.FindNext	23
6.5	pdf.FindPrev	23
6.6	pdf.FindStart	24
6.7	pdf.FreePage	25

6.8	pdf.GetBookmarks	25
6.9	pdf.GetBoundedText	26
6.10	pdf.GetBrush	27
6.11	pdf.GetBrushFromPage	28
6.12	pdf.GetCharBox	29
6.13	pdf.GetCharIndexAtPos	30
6.14	pdf.GetCharOrigin	31
6.15	pdf.GetCropBox	31
6.16	pdf.GetFindResult	32
6.17	pdf.GetLastError	32
6.18	pdf.GetMediaBox	33
6.19	pdf.GetMetaText	34
6.20	pdf.GetObjectType	34
6.21	pdf.GetPageLabel	35
6.22	pdf.GetPageLen	36
6.23	pdf.GetPageLinks	36
6.24	pdf.GetRects	37
6.25	pdf.GetText	38
6.26	pdf.GetVersion	39
6.27	pdf.IsPDF	39
6.28	pdf.LoadPage	39
6.29	pdf.OpenDocument	40
6.30	pdf.PageToDevice	41
7	Anmerkungsmethoden	43
7.1	annot:SetBorderStyle	43
7.2	annot:SetCMYKColor	44
7.3	annot:SetFreeTextAnnot2PointCalloutLine	44
7.4	annot:SetFreeTextAnnot3PointCalloutLine	45
7.5	annot:SetFreeTextAnnotDefaultStyle	45
7.6	annot:SetFreeTextAnnotLineEndingStyle	46
7.7	annot:SetGrayColor	47
7.8	annot:SetLineAnnotCaption	48
7.9	annot:SetLineAnnotLeader	48
7.10	annot:SetLineAnnotPosition	49
7.11	annot:SetLinkAnnotBorderStyle	50
7.12	annot:SetLinkAnnotHighlightMode	50
7.13	annot:SetMarkupAnnotCloudEffect	51
7.14	annot:SetMarkupAnnotCreationDate	51
7.15	annot:SetMarkupAnnotIntent	52
7.16	annot:SetMarkupAnnotInteriorCMYKColor	53
7.17	annot:SetMarkupAnnotInteriorGrayColor	53
7.18	annot:SetMarkupAnnotInteriorRGBColor	54
7.19	annot:SetMarkupAnnotInteriorTransparent	54
7.20	annot:SetMarkupAnnotPopup	55
7.21	annot:SetMarkupAnnotQuadPoints	55
7.22	annot:SetMarkupAnnotRectDiff	56
7.23	annot:SetMarkupAnnotSubject	56

7.24	annot:SetMarkupAnnotTitle	57
7.25	annot:SetMarkupAnnotTransparency	57
7.26	annot:SetNoColor	58
7.27	annot:SetPopupAnnotOpened	58
7.28	annot:SetRGBColor	59
7.29	annot:SetTextAnnotIcon	59
7.30	annot:SetTextAnnotOpened	60
8	Anpassungsmethoden	61
8.1	dest:SetFit	61
8.2	dest:SetFitB	61
8.3	dest:SetFitBH	61
8.4	dest:SetFitBV	62
8.5	dest:SetFitH	62
8.6	dest:SetFitR	63
8.7	dest:SetFitV	64
8.8	dest:SetXYZ	64
9	Bildmethoden	65
9.1	image:AddSMask	65
9.2	image:GetBitsPerComponent	65
9.3	image:GetColorSpace()	65
9.4	image:GetHeight	66
9.5	image:GetSize	66
9.6	image:GetWidth	66
9.7	image:SetColorMask	67
9.8	image:SetMaskImage	67
10	Dokumentmethoden	69
10.1	doc:AddPage	69
10.2	doc:AddPageLabel	69
10.3	doc:AttachFile	70
10.4	doc:CreateExtGState	71
10.5	doc:CreateImageFromBrush	71
10.6	doc:CreateImageFromMem	72
10.7	doc:CreateOutline	73
10.8	doc:Free	73
10.9	doc:GetCurrentEncoder	74
10.10	doc:GetCurrentPage	74
10.11	doc:GetEncoder	75
10.12	doc:GetError	75
10.13	doc:GetErrorDetail	76
10.14	doc:GetFont	76
10.15	doc:GetInfoAttr	77
10.16	doc:GetPageByIndex	77
10.17	doc:GetPageLayout	78
10.18	doc:GetPageMode	78

10.19	doc:GetViewerPreference	79
10.20	doc:InsertPage	79
10.21	doc:LoadFont	80
10.22	doc:LoadJPEGImage	81
10.23	doc:LoadPNGImage	81
10.24	doc:LoadRawImage	82
10.25	doc:LoadTTFont	83
10.26	doc:LoadType1Font	84
10.27	doc:ResetError	85
10.28	doc:SaveToFile	85
10.29	doc:SetCompressionMode	86
10.30	doc:SetCurrentEncoder	87
10.31	doc:SetEncryptionMode	87
10.32	doc:SetInfoAttr	88
10.33	doc:SetInfoDateAttr	89
10.34	doc:SetOpenAction	90
10.35	doc:SetPageLayout	90
10.36	doc:SetPageMode	91
10.37	doc:SetPagesConfiguration	92
10.38	doc:SetPassword	92
10.39	doc:SetPermission	93
10.40	doc:SetViewerPreference	94
10.41	doc:UseCNSEncodings	95
10.42	doc:UseCNSFonts	95
10.43	doc:UseCNTEncodings	96
10.44	doc:UseCNTFonts	97
10.45	doc:UseJPEncodings	97
10.46	doc:UseJPFonts	98
10.47	doc:UseKREncodings	99
10.48	doc:UseKRFonts	100
10.49	doc:UseUTFEncodings	101
11	Grafikzustandsmethoden	103
11.1	extgs:SetAlphaFill	103
11.2	extgs:SetAlphaStroke	103
11.3	extgs:SetBlendMode	104
12	Kodiermethoden	105
12.1	encoder:GetByteType	105
12.2	encoder:GetType	105
12.3	encoder:GetUnicode	105
12.4	encoder:GetWritingMode	106

13	Schriftartmethoden	107
13.1	font:GetAscent	107
13.2	font:GetBBox	107
13.3	font:GetCapHeight	107
13.4	font:GetDescent	108
13.5	font:GetEncodingName	108
13.6	font:GetFontName	109
13.7	font:GetUnicodeWidth	109
13.8	font:GetXHeight	109
13.9	font:MeasureText	110
13.10	font:TextWidth	111
14	Seitenmethoden	113
14.1	page:Arc	113
14.2	page:BeginText	113
14.3	page:Circle	114
14.4	page:Clip	114
14.5	page:ClosePath	115
14.6	page:ClosePathEofillStroke	115
14.7	page:ClosePathFillStroke	115
14.8	page:ClosePathStroke	116
14.9	page:Concat	116
14.10	page:CreateCircleAnnot	117
14.11	page:CreateDestination	118
14.12	page:CreateFreeTextAnnot	118
14.13	page:CreateHighlightAnnot	119
14.14	page:CreateLineAnnot	119
14.15	page:CreateLinkAnnot	120
14.16	page:CreatePopupAnnot	120
14.17	page:CreateProjectionAnnot	121
14.18	page:CreateSquareAnnot	122
14.19	page:CreateSquigglyAnnot	122
14.20	page:CreateStampAnnot	123
14.21	page:CreateStrikeOutAnnot	124
14.22	page:CreateTextAnnot	124
14.23	page:CreateTextMarkupAnnot	125
14.24	page:CreateUnderlineAnnot	126
14.25	page:CreateURILinkAnnot	127
14.26	page:CreateWidgetAnnot	127
14.27	page:CurveTo	128
14.28	page:CurveTo2	128
14.29	page:CurveTo3	129
14.30	page:DrawImage	129
14.31	page:Ellipse	130
14.32	page:EndPath	130
14.33	page:EndText	130
14.34	page:EoClip	131
14.35	page:Eofill	131

14.36	page:EofillStroke	132
14.37	page:ExecuteXObject	132
14.38	page:Fill	132
14.39	page:FillStroke	133
14.40	page:GetCharSpace	133
14.41	page:GetCMYKFill	133
14.42	page:GetCMYKStroke	134
14.43	page:GetCurrentFont	134
14.44	page:GetCurrentFontSize	135
14.45	page:GetCurrentPos	135
14.46	page:GetCurrentTextPos	136
14.47	page:GetDash	136
14.48	page:GetFillingColorSpace	137
14.49	page:GetFlat	137
14.50	page:GetGMode	137
14.51	page:GetGrayFill	138
14.52	page:GetGrayStroke	138
14.53	page:GetGStateDepth	139
14.54	page:GetHeight	139
14.55	page:GetHorizontalScaling	139
14.56	page:GetLineCap	140
14.57	page:GetLineJoin	140
14.58	page:GetLineWidth	140
14.59	page:GetMiterLimit	141
14.60	page:GetRGBFill	141
14.61	page:GetRGBStroke	142
14.62	page:GetStrokingColorSpace	142
14.63	page:GetTextLeading	143
14.64	page:GetTextMatrix	143
14.65	page:GetTextRenderingMode	144
14.66	page:GetTextRise	144
14.67	page:GetTransMatrix	144
14.68	page:GetWidth	145
14.69	page:GetWordSpace	145
14.70	page:GRestore	146
14.71	page:GSave	146
14.72	page:LineTo	147
14.73	page:MeasureText	147
14.74	page:MoveTextPos	148
14.75	page:MoveTo	148
14.76	page:MoveToNextLine	149
14.77	page:Rectangle	149
14.78	page:SetCharSpace	149
14.79	page:SetCMYKFill	150
14.80	page:SetCMYKStroke	150
14.81	page:SetDash	151
14.82	page:SetExtGState	151
14.83	page:SetFlat	152

14.84	page:SetFontAndSize	152
14.85	page:SetGrayFill	152
14.86	page:SetGrayStroke	153
14.87	page:SetHeight	153
14.88	page:SetHorizontalScaling	153
14.89	page:SetLineCap	154
14.90	page:SetLineJoin	154
14.91	page:SetLineWidth	155
14.92	page:SetMiterLimit	155
14.93	page:SetRGBFill	156
14.94	page:SetRGBStroke	156
14.95	page:SetRotate	156
14.96	page:SetSize	157
14.97	page:SetSlideShow	158
14.98	page:SetTextLeading	159
14.99	page:SetTextMatrix	159
14.100	page:SetTextRenderingMode	160
14.101	page:SetTextRise	161
14.102	page:SetWidth	161
14.103	page:SetWordSpace	161
14.104	page:SetZoom	162
14.105	page:ShowText	162
14.106	page:ShowTextNextLine	162
14.107	page:Stroke	163
14.108	page:TextOut	163
14.109	page:TextRect	164
14.110	page:TextWidth	165
15	Umrissmethoden	167
15.1	outline:SetDestination	167
15.2	outline:SetOpened	167
Anhang A	Lizenzen	169
A.1	LibHaru license	169
A.2	LuaHPDF license	169
A.3	PDFium license	169
Index		171

1 Allgemeine Informationen

1.1 Einführung

Polybios ist ein Plugin für Hollywood, mit dem Sie auf einfache Weise PDF-Dokumente aus Hollywood-Skripten erstellen können. Darüber hinaus kann Polybios auch bestehende PDF-Dokumente öffnen und deren Seiten in Hollywood-Pinsel umwandeln. Bei der Konvertierung von PDF-Seiten in Hollywood-Pinsel erstellt Polybios für Sie Vektorpinsel, die ohne Qualitätsverlust skaliert, gedreht und transformiert werden können (es sei denn, Bitmap-Grafiken sind in das PDF-Dokument eingebettet).

Polybios bietet über 200 Funktionen zur Erstellung von PDF-Dokumenten aller Art. Es unterstützt Grafik-Grundelemente, Text in verschiedenen Kodierungen einschließlich Unicode, Einbettung von Schriftarten sowie Bilder und Hollywood-Pinsel in PDF-Dokumente. Darüber hinaus unterstützt Polybios die Erstellung von passwortgeschützten PDF-Dokumenten, verschlüsselten PDF-Dokumenten, Komprimierung, Dateianhängen, Anmerkungen, erweiterte Grafiken, Info-Wörterbüchern, RGB-, CMYK- und Graufarbräumen, verschiedene Anzeigemodi, Übergangseffekten, Links und Berechtigungsflags für PDF-Dokumente. Auch die Transformation von PDF-Objekten wird vollständig unterstützt. Schließlich kann Polybios auch PDF-Dokumente mit einer einfach zu navigierenden Gliederung erstellen, die auch als Inhaltsverzeichnis verwendet werden kann.

Polybios unterstützt auch das Extrahieren von Text aus PDF-Seiten, das Abrufen aller Lesezeichen in einem Dokument, die Handhabung von Links auf PDF-Seiten und die Suche nach Seiten. Darüber hinaus ermöglicht Polybios die Abfrage der Position von Text auf PDF-Seiten, so dass z.B. Textmarkierungsfunktionen implementiert werden können.

Polybios enthält eine umfangreiche Dokumentation in verschiedenen Formaten wie PDF (natürlich), HTML, AmigaGuide und CHM, welche detaillierte Beschreibungen aller Befehle und Methoden des Plugins enthält. Darüber hinaus sind über 25 Beispielskripte im Distributionsarchiv enthalten, um Ihnen einen schnellen Einstieg zu ermöglichen.

All dies macht Polybios zum ultimativen PDF-Tool für Hollywood, das alles enthält, was Sie brauchen, um Geschichte in PDF zu schreiben!

1.2 Lizenzvereinbarung

Polybios ist © Copyright 2013-2020 bei Andreas Falkenhahn (im folgenden "der Autor" genannt). Alle Rechte vorbehalten.

Das Programm wird zur Verfügung gestellt "wie es ist" und der Autor kann für keinerlei Schäden, welcher Natur sie auch immer sein mögen, verantwortlich gemacht werden. Sie benutzen dieses Programm völlig auf eigene Gefahr und eigenes Risiko. Der Autor gibt keinerlei Garantien in Verbindung mit der Benutzung dieses Programmes, nicht einmal die Garantie der Funktionstüchtigkeit.

Dieses Plugin kann frei weitergegeben werden solange die folgenden drei Bedingungen erfüllt sind:

1. Es dürfen keine Änderungen am Programm vorgenommen werden.
2. Das Programm darf nicht verkauft werden.

3. Wenn Sie das Programm auf einer Coverdisk veröffentlichen möchten, müssen Sie erst um Erlaubnis fragen.

Dieses Programm benutzt LibHaru von Takeshi Kanno und Antony Dovgal. Siehe [Abschnitt A.1 \[LibHaru-Lizenz\]](#), Seite 169, für Details.

Dieses Programm benutzt LuaHPDF von Kurt Jung. Siehe [Abschnitt A.2 \[LuaHPDF-Lizenz\]](#), Seite 169, für Details.

Dieses Programm benutzt PDFium von dem PDFium-Team. Siehe [Abschnitt A.3 \[PDFium-Lizenz\]](#), Seite 169, für Details.

Alle Warenzeichen sind Eigentum ihrer jeweiligen Firmen.

FÜR DIESES PROGRAMM GIBT ES KEINE GARANTIE, SOWEIT ES DIE ANZUWENDENDEN GESETZE ZULASSEN. SOFERN ANDERSWO NICHTS GEGENTEILIGES GESCHRIEBEN STEHT STELLEN DER AUTOR UND/ODER DRITTE DAS PROGRAMM "SO WIE ES IST" ZUR VERFÜGUNG, OHNE IRGEND-EINE GARANTIE, WEDER DIREKT NOCH INDIREKT. DIES BEINHÄLTET, IST ABER NICHT DARAUF BESCHRÄNKT, VERKÄUFLICHKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN VERWENDUNGSZWECK. DAS VOLLSTÄNDIGE RISIKO DER QUALITÄT UND AUSFÜHRBARKEIT DES PROGRAMMS LIEGT BEIM ANWENDER. SOLLTE SICH DAS PROGRAMM ALS DEFEKT HERAUSSTELLEN, LIEGEN ALLE KOSTEN FÜR SERVICE, INSTANDSETZUNG ODER NACHBESSERUNG BEIM ANWENDER.

KEIN COPYRIGHT-INHABER ODER DRITTER, DER DAS PROGRAMM WIE OBEN ERLAUBT WEITERVERKAUFT, KANN FÜR SCHÄDEN IRGENDWELCHER ART HAFTBAR GEMACHT WERDEN (DIES BEINHÄLTET, IST ABER NICHT BESCHRÄNKT AUF, DATENVERLUST INFOLGE UNFÄHIGKEIT DES PROGRAMMS, MIT ANDEREN PROGRAMMEN ZUSAMMENZUARBEITEN), SELBST WENN EIN SOLCHER INHABER ODER DRITTER AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WURDE, AUSSER ES BESTEHT EINE SCHRIFTLICHE EINWILLIGUNG ODER WIRD VOM GESETZ VERLANGT.

1.3 Voraussetzung

- Hollywood 7.1 oder besser
- Unter macOS benötigt Polybios mindestens 10.9 auf x86 und x64 Systemen und 10.5 auf PowerPC Systemen
- Auf Android ist mindestens Version 5.0 erforderlich.
- Wenn Sie WinUAE verwenden, benötigen Sie mindestens WinUAE 4.2.1 oder Polybios kann wegen eines Fehlers in der 68020-Emulation von WinUAE abstürzen.

1.4 Installation

Die Installation von Polybios ist unkompliziert und einfach: Kopieren Sie einfach die Datei `polybios.hwp` für die Plattform Ihrer Wahl in Hollywoods Plugin-Verzeichnis. Auf allen Systemen außer auf AmigaOS und kompatiblen müssen Plugins in einem Verzeichnis mit dem Namen `Plugins` gespeichert werden, das sich im selben Verzeichnis wie das Hauptprogramm von Hollywood befindet. Auf AmigaOS und kompatiblen Systemen

müssen Plugins stattdessen in `LIBS:Hollywood` installiert werden. Unter macOS X muss sich das Verzeichnis `Plugins` im Verzeichnis `Resources` des Programmpakets befinden, d.h. im Verzeichnis `HollywoodInterpreter.app/Contents/Resources`. Beachten Sie, dass `HollywoodInterpreter.app` im Programmpaket `Hollywood.app` selbst gespeichert ist, nämlich in `Hollywood.app/Contents/Resources`.

Anschliessend kopieren Sie den Inhalt des Ordners `Examples` in den Ordner `Examples` Ihrer Hollywood-Installation. Alle Polybios-Beispiele erscheinen dann in Hollywoods GUI und Sie können sie bequem von der Hollywood-GUI oder IDE aus starten und anzeigen.

Unter Windows sollten Sie auch die Datei `Polybios.chm` in das Verzeichnis `Docs` Ihrer Hollywood-Installation kopieren. Wenn sich dann der Cursor über einem Polybios-Befehl in der Hollywood-IDE befindet, können Sie die Online-Hilfe aufrufen, indem Sie F1 drücken.

Unter Linux und macOS kopieren Sie das Verzeichnis `Polybios`, das sich im Verzeichnis `Docs` des Polybios-Distributionsarchivs befindet, in das Verzeichnis `Docs` Ihrer Hollywood-Installation. Beachten Sie, dass sich unter macOS das Verzeichnis `Docs` innerhalb des Programmpakets `Hollywood.app` befindet, d.h. in `Hollywood.app/Contents/Resources/Docs`.

2 Über Polybios

2.1 Danksagungen

Polybios wurde von Andreas Falkenhahn geschrieben, basierend auf der Arbeit von Takeshi Kanno, Antony Dovgal, Kurt Jung und den PDFium-Autoren. Besonderer Dank gilt Sebastian Bauer, der clib2 um rudimentäre Wide Character-Unterstützung erweitert hat, so dass PDFium auch auf AmigaOS 4 kompiliert werden kann. Ein weiterer Dank geht an Stefan "Bebbo" Franke für die Pflege einer aktuellen Version von gcc, die für die Motorola 680x0 Serie kompilieren kann.

Ein besonderer Dank geht an Dominic Widmer und Helmut Haake für die Übersetzung des Handbuchs ins Deutsche. Fehler oder Verbesserungsvorschläge bzgl. des deutschen Handbuchs bitte an das Übersetzungsteam richten, welches unter handbuch@gmx.ch oder <https://amiga-resistance.info> erreicht werden kann.

Wenn Sie mich kontaktieren möchten, senden Sie bitte eine E-Mail an andreas@airsoftsoftwair.de oder nutzen Sie das Kontaktformular unter <http://www.hollywood-mal.com>.

2.2 Häufig gestellte Fragen

In diesem Abschnitt werden einige häufig gestellte Fragen behandelt. Bitte lesen Sie sie zuerst, bevor Sie in der Mailingliste oder im Forum nachfragen, da Ihr Problem hier möglicherweise behandelt wurde.

F: Wie kann ich bestehende PDF-Dokumente ändern?

A: Das wird derzeit nicht unterstützt, ist aber für eine zukünftige Version von Polybios geplant.

F: Warum unterstützt Polybios die Konvertierung von PDF-Seiten in Vektorpinsel AROS nicht?

A: Das liegt daran, dass PDFium einen Compiler benötigt, der in der Lage ist, C++11 und breite Zeichen zu verarbeiten, was derzeit für AROS nicht verfügbar ist. Aber das wird sich hoffentlich in Zukunft ändern, so dass auch AROS-Anwender PDF-Seiten in Hollywood-Pinsel umwandeln können.

F: Warum werden Chinesische/Japanische/Koreanische (CJK) Zeichen in meinem Dokument nicht korrekt dargestellt?

A: Stellen Sie sicher, dass Sie eine TrueType-Schriftart mit CJK-Unterstützung installiert haben. Installieren Sie beispielsweise Konatu auf Ihrem System und CJK-Zeichen sollten korrekt dargestellt werden.

F: Die 68k-Version von Polybios funktioniert nicht unter der OS4-Emulation.

A: Es scheint, dass der OS4-JIT Probleme mit dem PDF-Renderer von Polybios hat, der ein 5 MB großes Binary ist. Wenn Sie die 68k-Version von Polybios auf OS4 wirklich verwenden

wollen, müssen Sie den JIT für die Datei LIBS:Hollywood/Polybios.ext deaktivieren. Dann sollte es funktionieren.

F: Gibt es ein Hollywood-Forum, in dem ich mit anderen Benutzern in Kontakt treten kann?

A: Ja, bitte besuchen Sie die "Community" oder "Forum"-Sektion des offiziellen Hollywood-Portals unter <http://www.hollywood-mal.com>.

F: Wo kann ich um Hilfe bitten?

A: Es gibt ein lebhaftes englischsprachiges Forum auf <http://forums.hollywood-mal.com> und wir haben auch eine Mailing-Liste (auch englischsprachig), auf die Sie unter airsoft_hollywood@yahoo.com zugreifen können. Besuchen Sie <https://www.hollywood-mal.de/> für Informationen darüber, wie Sie der Mailingliste beitreten können. Ausserdem ist ein deutschsprachiges Forum vorhanden, welches Sie unter <https://www.amiga-resistance.info/> erreichen können.

F: Ich habe einen Fehler gefunden.

A: Bitte informieren Sie mich darüber in den speziellen Bereichen des Forums oder der Mailingliste.

2.3 Zukunft

Hier sind einige Punkte, die auf meiner Aufgabenliste stehen:

- Unterstützung für die Wiedergabe von PDF-Seiten auf AROS
- Unterstützung für die Bearbeitung vorhandener PDF-Dokumente

Zögern Sie nicht, mich zu kontaktieren, wenn Polybios eine bestimmte Funktion fehlt, die für Ihr Projekt wichtig ist.

2.4 Geschichte

Bitte schauen Sie in die auf englisch verfasste Datei `history.txt`. Hier finden Sie ein vollständiges Änderungsprotokoll von Polybios.

3 Anzeigen von PDFs

3.1 Übersicht

Es gibt zwei verschiedene Methoden, um PDF-Dokumente mit Polybios in Hollywood zu betrachten: Sie können entweder einzelne PDF-Seiten als Vektorpinsel oder ein ganzes PDF-Dokument als Hollywood-Animation laden. Die Seiten des Dokuments werden dann einfach auf einzelne Animationsbilder abgebildet. Welchen Weg Sie auch wählen, Polybios bildet PDF-Seiten immer auf Vektorobjekte in Hollywood ab, so dass sie ohne Qualitätsverluste skaliert, gedreht und transformiert werden können.

Um Polybios von Ihrem Hollywood-Skript aus zu verwenden, müssen Sie zunächst das Plugin am Anfang Ihres Skripts mit der folgenden Zeile initialisieren:

```
@REQUIRE "polybios"
```

Es gibt auch einige zusätzliche Argumente, die Sie an den Präprozessorbefehl `@REQUIRE` übergeben können. Die folgenden Argumente sind derzeit verfügbar:

`NoVectorAnim`:

Wenn Sie PDFs als Hollywood-Anim laden, wird Polybios automatisch Vektor-Animationen erzeugen, wenn die mit Polybios verwendete Hollywood-Version dies unterstützt. Wenn Sie das nicht wollen, setzen Sie diesen Tag auf `True`. Beachten Sie, dass es beim Skalieren oder Transformieren der Animation, die die PDF-Seiten enthält, zu Qualitätsverlusten kommt, wenn Sie diesen Tag auf `True` setzen. Der Standardwert ist `False`. (V1.3)

3.2 Laden von Seiten als Vektorpinsel

Um PDF-Seiten als Vektorpinsel zu laden, müssen Sie das PDF-Dokument mit dem Befehl `pdf.OpenDocument()` öffnen und dann die gewünschten Seiten mit dem Befehl `pdf.GetBrush()` in Hollywood-Vektorpinsel konvertieren.

Hier ist ein Beispiel:

```
pdf.OpenDocument(1, "test.pdf")
pdf.GetBrush(1, 1, 1)
DisplayBrush(1, #CENTER, #CENTER)
FreeBrush(1)
pdf.CloseDocument(1)
```

Der obige Code öffnet das PDF-Dokument mit dem Namen `test.pdf` und konvertiert seine erste Seite in einen Vektorpinsel. Es zeigt dann diesen Vektorpinsel in der Mitte des Displays. Beachten Sie, dass der Vektorpinsel immer noch vom PDF-Dokument abhängig ist. Daher ist es nicht erlaubt, den Befehl `pdf.CloseDocument()` im Dokument aufzurufen, solange Sie den Pinsel noch benötigen. Deshalb müssen wir zuerst den Pinsel aus dem Speicher löschen und schließen anschließend das Dokument. Andernfalls wird ein Fehler angezeigt.

Sie können die Anzahl der Seiten im PDF-Dokument ermitteln, indem Sie zuerst den Objekttyp für PDF-Dokumente ermitteln und dann den Befehl `GetAttribute()` von Hollywood verwenden:

```
PDF_DOCUMENT = pdf.GetObjectType()
```

```
numpages = GetAttribute(PDF_DOCUMENT, 1, #PDFATTRPAGES)
```

Der obige Code ermittelt die Anzahl der Seiten aus dem PDF-Dokument mit dem Identifikator 1 und speichert sie in der Variable `numpages`.

3.3 Laden von Seiten als Animation

Alternativ bietet Polybios an, ein komplettes PDF-Dokument in ein Hollywood-Animationsobjekt zu laden. Sie können dann auf die einzelnen Seiten zugreifen, indem Sie einfach die Einzelbilder der Animation aufrufen.

So laden Sie ein PDF-Dokument als Hollywood-Animation:

```
LoadAnim(1, "test.pdf", {FromDisk = True})
For Local k = 1 To GetAttribute(#ANIM, 1, #ATTRNUMFRAMES)
    DisplayAnimFrame(1, #CENTER, #CENTER, k)
    WaitLeftMouse
Next
```

Der obige Code zeigt alle Seiten eines PDF-Dokuments. Sie müssen die linke Maustaste drücken, um zur nächsten Seite zu springen.

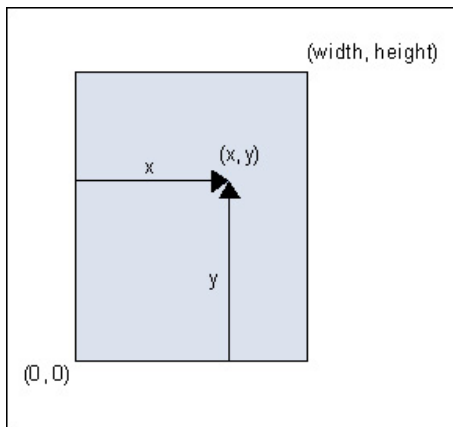
Beachten Sie, dass wir in unserem Aufruf von `LoadAnim()` den Tag `FromDisk` auf `True` setzen. Dies ist sehr wichtig, da sonst alle PDF-Seiten in den Speicher geladen und gepuffert werden, was bei größeren PDF-Dokumenten eine große Verschwendung von Speicher sein kann.

Sie können das PDF-Dokument natürlich auch mit der Präprozessor-Anweisung `@ANIM` anstelle von `LoadAnim()` laden.

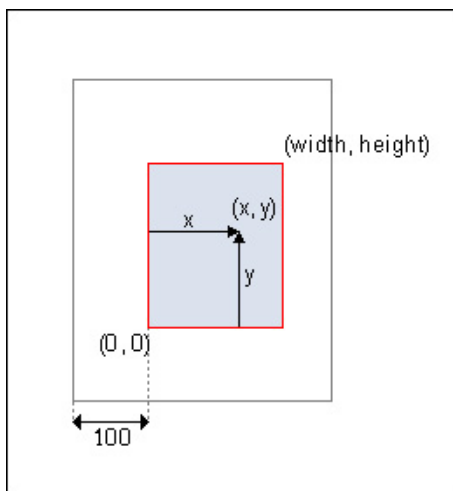
4 Erstellen von PDFs

4.1 Koordinatensystem

Beachten Sie, dass PDF-Dokumente ein anderes Koordinatensystem als Hollywood verwenden. Im unten gezeigten Standard-Koordinatensystem von PDF befindet sich die linke untere Ecke auf den Koordinaten $(0, 0)$ und die rechte obere Ecke auf den Koordinaten (Breite, Höhe). Die Standardauflösung beträgt 72dpi. In Hollywood ist die obere linke Ecke bei $(0, 0)$.



Ein Programm kann das Koordinatensystem durch Aufruf von `page:Concat()` ändern. Wenn ein Programm beispielsweise `page:Concat(0.5, 0, 0, 0.5, 100, 100)` im Standardformat aufruft, wird das oben gezeigte Koordinatensystem in das in der folgenden Abbildung gezeigte neue System umgewandelt:

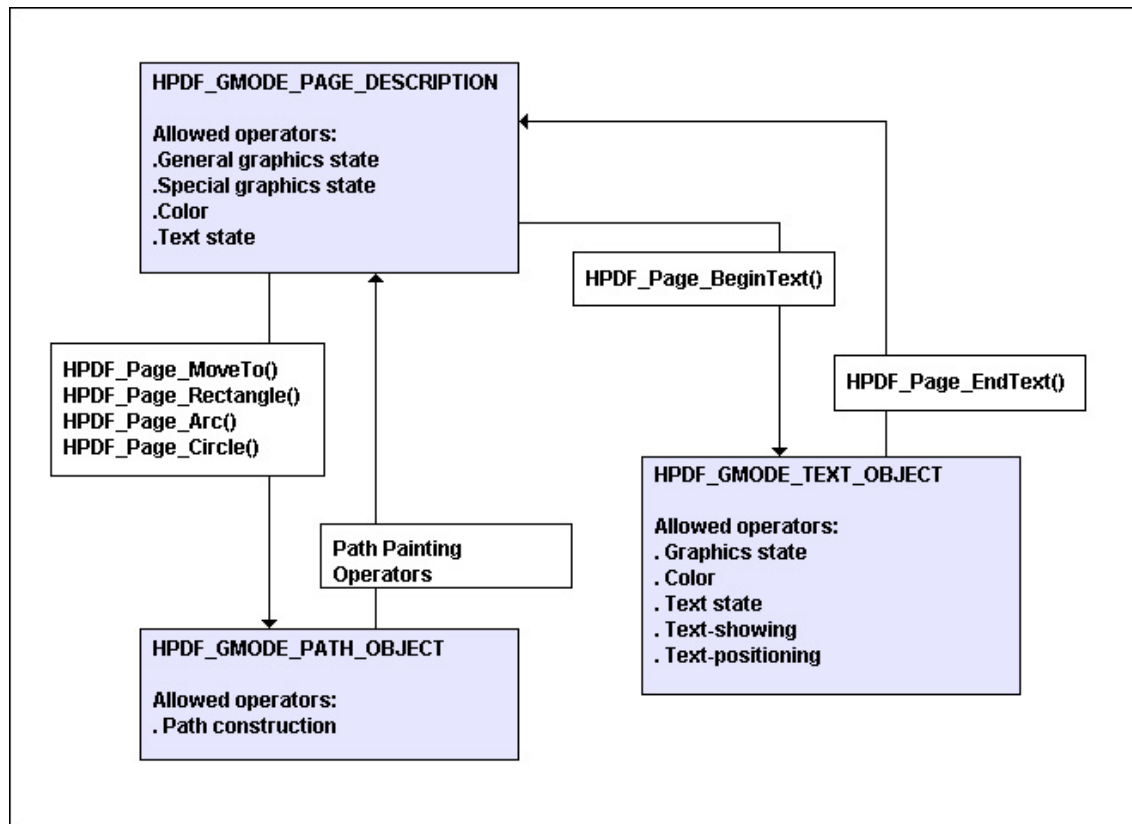


4.2 Grafikmodus

In Polybios verwaltet jedes Seitenobjekt ein Flag namens "graphics mode". Der Grafikmodus entspricht dem Grafikobjekt der PDF-Spezifikation.

Der Grafikmodus wird durch das Aufrufen bestimmter Befehle geändert. Die Befehle, die aufgerufen werden können, werden durch den Wert des Grafikmodus entschieden.

Die folgende Abbildung zeigt die Beziehungen der Grafikmodi.



4.3 Pfade zeichnen

Ein Pfad besteht aus geraden und gebogenen Liniensegmenten. Pfade definieren Formen und Regionen.

Vektorgrafiken werden in den folgenden Schritten gezeichnet:

1. Setzen Sie die Grafikzustände (z.B. Linienbreite, Strichmuster, Farbe...) mit Hilfe von Grafikstatus- oder Farb-Operatoren.
2. Starten Sie einen neuen Pfad mit `page:MoveTo()`, `page:Rectangle()`, `page:Arc()` oder `page:Circle()`.
3. Hängen Sie einen neuen Pfad mit Hilfe von Pfadkonstruktions-Operatoren an den bestehenden Pfad an.
4. Zeichnen Sie den Pfad mit Hilfe von Pfadzeichnungs-Operatoren.

Hier ist eine Liste der Grafikstatus-Operatoren:

```

page:Concat()
page:SetDash()
page:SetFlat()
page:SetLineCap()

```

```
page:SetLineJoin()  
page:SetLineWidth()  
page:SetMiterLimit()
```

Hier ist eine Liste der Farb-Operatoren:

```
page:SetCMYKFill()  
page:SetCMYKStroke()  
page:SetGrayFill()  
page:SetGrayStroke()  
page:SetRGBFill()  
page:SetRGBStroke()
```

Hier ist eine Liste der Pfadkonstruktions-Operatoren:

```
page:Arc()  
page:Circle()  
page:CurveTo()  
page:CurveTo2()  
page:CurveTo3()  
page:LineTo()  
page:MoveTo()  
page:Rectangle()
```

Hier ist eine Liste der Pfadzeichnungs-Operatoren:

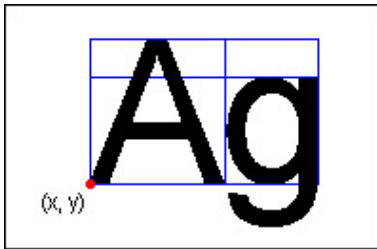
```
page:ClosePathFillStroke()  
page:ClosePathEofillStroke()  
page:ClosePathStroke()  
page:Eofill()  
page:EofillStroke()  
page:EndPath()  
page:Fill()  
page:FillStroke()  
page:Stroke()
```

4.4 Text ausgeben

Der Text wird durch die folgenden Schritte ausgegeben:

1. Beginnen Sie mit dem Ausgeben von Text durch Aufruf von `page:BeginText()`.
2. Setzen Sie die Textzustände (z.B. Schriftart, Füllfarbe....) mit Hilfe von Textstatus- oder Farb-Operatoren. Mindestens `page:SetFontAndSize()` muss einmal aufgerufen werden, bevor Text-Operatoren aufgerufen werden.
3. Setzen Sie die Textpositionierung durch Aufruf von Textpositionierungs-Operatoren.
4. Zeigen Sie Text durch den Aufruf von Textzeichnungs-Operatoren an.
5. Die Schritte 2 bis 4 ggf. wiederholen.
6. Beenden Sie das Ausgeben von Text durch Aufruf von `page:EndText()`.

Die folgende Abbildung verdeutlicht die Positionierung des Textes:



Sie sehen, dass im Gegensatz zum Hollywood-Koordinatensystem das Koordinatensystem des PDF-Dokuments für die Platzierung von Text unten beginnt und sich nach oben erstreckt.

Hier ist eine Liste der Textstatus-Operatoren:

```
page:SetCharSpace()
page:SetFontAndSize()
page:SetHorizontalScaling()
page:SetTextLeading()
page:SetTextRenderingMode()
page:SetTextRise()
page:SetWordSpace()
```

Hier ist eine Liste von Textpositionierungs-Operatoren:

```
page:MoveTextPos()
page:SetTextMatrix()
```

Hier ist eine Liste von Textzeichnungs-Operatoren:

```
page:ShowText()
page:ShowTextNextLine()
page:TextOut()
page:TextRect()
```

4.5 Farben

Farben werden durch drei reelle Zahlen (d.h. solche mit Dezimalpunkt) in der Form R G B angegeben, wobei jede Zahl den Anteil von Rot (R), Grün (G) und Blau (B) in einer Farbe definiert. Die gültigen Zahlen sind von 0,0 bis einschließlich 1,0.

4.6 Schriftarten

Es gibt verschiedene Schriftarten in Polybios.

Base14 font:

Die eingebaute Schriftart von PDF. Verfügbar in allen Anzeigeprogrammen.

Type1 font:

Ein von PostScript verwendetes Schriftformat.

TrueType font:

Weit verbreitetes Umrisschriftformat.

CID font: Schriftformat für Multi-Byte-Zeichen. Entwickelt von Adobe.

Hollywood-Skripte können `doc:GetFont()` verwenden, um mit Schriftarten zu arbeiten. Zuvor muss einer der folgenden Befehle verwendet werden, um die Schriftart vor dem Aufruf von `doc:GetFont()` zu laden (außer Base14-Fonts; diese sind immer verfügbar und müssen nicht geladen werden):

```
HPDF_LoadType1FontFromFile()
HPDF_LoadTTFontFromFile()
HPDF_LoadTTFontFromFile2()
HPDF_UseCNSFonts()
HPDF_UseCNTFonts()
HPDF_UseJPFonts()
HPDF_UseKRFonts()
```

4.7 Base14-Schriftarten

Base14-Schriftarten sind im PDF integriert und können von allen Anzeigeprogramme dargestellt werden. Ein Programm kann jederzeit eine Base14-Schriftart benutzen, indem es `doc:GetFont()` aufruft. PDF-Dateien, die Base14-Schriftarten verwenden, sind kleiner als solche, die andere Schriftarten verwenden. Außerdem ist die PDF-Verarbeitung schneller, da keine externen Schriftarten geladen werden müssen. Base14-Schriftarten können jedoch nur den Latin-1-Zeichensatz anzeigen. Um andere Zeichensätze zu verwenden, muss ein Programm andere Schriftarten verwenden.

Die folgenden sind eingebaute Base14-Schriftarten. Sie sind in jedem PDF-Anzeigeprogramm verfügbar:

```
Courier
Courier-Bold
Courier-Oblique
Courier-BoldOblique
Helvetica
Helvetica-Bold
Helvetica-Oblique
Helvetica-BoldOblique
Times-Roman
Times-Bold
Times-Italic
Times-BoldItalic
Symbol
ZapfDingbats
```

4.8 Type1-Schriftarten

Type1 ist ein Format von Umrisschriftarten, die von Adobe entwickelt wurden. Eine AFM-Datei ist erforderlich, um eine externe Type1-Schrift mit Polybios zu verwenden. Wenn ein Hollywood-Skript eine externe Type1-Schriftart verwendet, muss es `doc:LoadType1Font()` aufrufen, bevor `doc:GetFont()` aufgerufen wird. Der Rückgabewert von `doc:LoadType1Font()` wird als Parameter für den Schriftnamen von `doc:GetFont()`

verwendet. Wenn beim Aufruf von `doc:LoadType1Font()` eine PFA/PFB-Datei angegeben wird, werden die Glyphendaten der Schriftart in die PDF-Datei eingebettet. Andernfalls werden nur Metrikdaten in der AFM-Datei eingefügt.

Hier ist ein Beispiel:

```
fontname = doc:LoadType1Font("a0100131.afm", "a0100131.pfb")
hfont = doc:GetFont(fontname, "CP1250")
page:SetFontAndSize(hfont, 10.5)
```

4.9 TrueType-Schriftarten

Polybios kann TrueType-Schriftarten verwenden, wovon es zwei Arten gibt: Das erste Format enthält nur eine Schriftart in seiner Datei und verwendet die Erweiterung ".ttf". Das zweite Format, welches die Erweiterung ".ttc" verwendet, enthält mehrere Schriftarten in seiner Datei. Deshalb hat `doc:LoadTTFont()` einen Parameter, mit dem der Index der zu ladenden Schriftart angegeben wird. Wenn der zusätzliche Parameter `embedding` beim Aufruf von `doc:LoadTTFont()` auf `True` gesetzt ist, wird die Untermenge der Schriftart in die PDF-Datei eingebettet. Wenn nicht, werden nur die Matrix-Daten in der PDF-Datei gespeichert. In diesem Fall kann ein Anzeigeprogramm eine alternative Schriftart verwenden, wenn sie die Schriftart nicht findet.

Hier ist ein Beispiel:

```
fontname = doc:LoadTTFont("arial.ttf", True)
hfont = doc:GetFont(fontname, "CP1250")
page:SetFontAndSize(hfont, 10.5)
```

Beachten Sie, dass Polybios nur TrueType-Schriftarten verwenden kann, die eine Unicode-Cmap und eine der folgenden Tabellen haben: "OS/2", "cmap", "cvt ", "fpgm", "glyph", "head", "hhea", "hmtx", "loca", "maxp", "name", "post", "prep".

4.10 CID-Schriftarten

CID-Schriftarten sind ein von Adobe entwickeltes Multi-Byte-Zeichensatz-Format. Zwei vereinfachte chinesische, eine traditionelle chinesische, vier japanische und vier koreanische Schriftarten sind in Polybios verfügbar. Hollywood-Skripte müssen die folgenden Befehle einmal aufrufen, bevor sie CID-Schriftarten verwenden können:

`doc:UseCNSFonts()`

Macht vereinfachte chinesische Schriftarten (SimSun, SimHei) verfügbar.

`doc:UseCNTFonts()`

Macht traditionelle chinesische Schriftarten (MingLiU) verfügbar.

`doc:UseJPFonts()`

Macht japanische Schriftarten (MS-Mincyo, MS-Gothic, MS-PMincyo, MS-PGothic) verfügbar.

`doc:UseKRFonts()`

Macht koreanische Schriftarten (Batang, Dotum, BatangChe, DotumChe) verfügbar.

Hier ist ein Beispiel:

```
doc:UseJPFonts()
```



```
doc:UseJPEncodings()  
hfont = doc:GetFont("MS-Mincyo", "90ms-RKSJ-H")  
page:SetFontAndSize(hfont, 10.5)
```

4.11 Kodierungen

Die folgenden Einzel-Byte-Kodierungen sind in Polybios verfügbar. Hollywood-Skripte können mit `doc:GetEncoder()` eine Kodier-Kennung ermitteln:

StandardEncoding

Die Standardkodierung von PDF

MacRomanEncoding

Die Standardkodierung von macOS

WinAnsiEncoding

Die Standardkodierung von Windows

FontSpecific

Verwendet die eingebaute Kodierung einer Schriftart

IS08859-2

Latin Alphabet Nr. 2 (Mitteleuropäisch)

IS08859-3

Latin Alphabet Nr. 3 (Südeuropäisch)

IS08859-4

Latin Alphabet Nr. 4 (Nordeuropäisch)

IS08859-5

Latin Kyrillisch Alphabet

IS08859-6

Latin Arabisch Alphabet

IS08859-7

Latin Griechisch Alphabet

IS08859-8

Latin Hebräisch Alphabet

IS08859-9

Latin Alphabet Nr. 5 (Türkisch)

IS08859-10

Latin Alphabet Nr. 6 (Nordisch)

IS08859-11

Thai, TIS 620-2569 Zeichensatz

IS08859-13

Latin Alphabet Nr. 7 (Baltisch)

IS08859-14

Latin Alphabet Nr. 8 (Keltisch)

- IS08859-15
Latin Alphabet Nr. 9 (Westeuropäisch)
- IS08859-16
Latin Alphabet Nr. 10 (Südosteuropäisch)
- CP1250 Microsoft Windows Codepage 1250 (Mitteleuropäisch)
- CP1251 Microsoft Windows Codepage 1251 (Kyrillisch)
- CP1252 Microsoft Windows Codepage 1252 (ANSI, Westeuropäisch)
- CP1253 Microsoft Windows Codepage 1253 (Griechisch)
- CP1254 Microsoft Windows Codepage 1254 (Türkisch)
- CP1255 Microsoft Windows Codepage 1255 (Hebräisch)
- CP1256 Microsoft Windows Codepage 1256 (Arabisch)
- CP1257 Microsoft Windows Codepage 1257 (Baltisch)
- CP1258 Microsoft Windows Codepage 1258 (Vietnamesisch)
- KOI8-R Russischer Netzzeichensatz

Die folgenden Multi-Byte-Kodierungen sind in Polybios verfügbar:

- GB-EUC-H EUC-CN Kodierung
- GB-EUC-V Vertikale Schreibversion von GB-EUC-H
- GBK-EUC-H
Microsoft Codepage 936 (IfCharSet 0x86) GBK Kodierung
- GBK-EUC-V
Vertikale Schreibversion von GBK-EUC-H
- ETen-B5-H
Microsoft Codepage 950 (IfCharSet 0x88) Big5 Zeichensatz mit ETen-Erweiterungen
- ETen-B5-V
Vertikale Schreibversion von ETen-B5-H
- 90ms-RKSJ-H
Microsoft Codepage 932, JIS X 0208 Schriftzeichen
- 90ms-RKSJ-V
Vertikale Schreibversion von 90ms-RKSJ-V
- 90msp-RKSJ-H
Microsoft Codepage 932, JIS X 0208 Schriftzeichen (proportional)
- EUC-H JIS X 0208 Zeichensatz, EUC-JP Kodierung
- EUC-V Vertikale Schreibversion von EUC-H
- KSC-EUC-H
KS X 1001:1992 Zeichensatz, EUC-KR Kodierung

KSC-EUC-V

Vertikale Schreibversion von KSC-EUC-V

KSCms-UHC-H

Microsoft Codepage 949 (lfCharSet 0x81), KS X 1001:1992 Zeichensatz plus 8822 zusätzlich Hangul, Unified Hangul Code (UHC) Kodierung (proportional)

KSCms-UHC-HW-H

Microsoft Codepage 949 (lfCharSet 0x81), KS X 1001:1992 Zeichensatz plus 8822 zusätzlich Hangul, Unified Hangul Code (UHC) Kodierung (feste Breite)

KSCms-UHC-HW-V

Vertikale Schreibversion von KSCms-UHC-HW-H

UTF-8 UTF-8 Kodierung.

Ein Hollywood-Skript muss eine der folgenden Befehle aufrufen, bevor es Multi-Byte-Kodierungen verwendet:

doc:UseCNSEncodings()

Macht vereinfachte chinesische Kodierungen (GB-EUC-H, GB-EUC-V, GBK-EUC-H, GBK-EUC-V) verfügbar.

doc:UseCNTEncodings()

Macht traditionelle chinesische Kodierungen (ETen-B5-H, ETen-B5-V) verfügbar.

doc:UseJPEncodings()

Macht Japanische Kodierungen (90ms-RKSJ-H, 90ms-RKSJ-V, 90msp-RKSJ-H, EUC-H, EUC-V) verfügbar.

doc:UseKREncodings()

Macht koreanische Kodierungen (KSC-EUC-H, KSC-EUC-V, KSCms-UHC-H, KSCms-UHC-HW-H, KSCms-UHC-HW-V) verfügbar.

doc:UseUTFEncodings()

Macht UTF-8 Kodierung verfügbar.

5 Tutorial

5.1 Tutorial

In diesem Tutorial erfahren Sie, wie Sie mit Polybios Ihr erstes PDF-Dokument erstellen. Das PDF-Dokument enthält zwei Seiten, eine mit einem Kreis und eine mit dem Text "Hallo Welt".

Zunächst müssen Sie ein Dokumentobjekt anlegen. Dies geschieht durch den Aufruf von `pdf.CreateDocument()`, das ein Dokumentobjekt für Sie erstellt. Das von `pdf.CreateDocument()` zurückgegebene Dokumentobjekt-Kennung wird dann in den folgenden Schritten verwendet.

```
doc = pdf.CreateDocument()
```

In einem zweiten Schritt können Sie einige Dokumentattribute setzen. Zum Beispiel setzen wir hier Komprimierung, Verschlüsselung, Seitenmodus und ein Passwort:

```
; Setzt den Kompressionsmodus
doc:SetCompressionMode(#HPDF_COMP_ALL)

; legt den Seitenmodus für die Verwendung von Umrissen fest
doc:SetPageMode(#HPDF_PAGE_MODE_USE_OUTLINE)

; Setzt das Passwort
doc:SetPassword("owner", "user")
```

Nachdem Sie die Dokumentattribute festgelegt haben, rufen Sie `doc:AddPage()` auf, um dem Dokument eine Seite hinzuzufügen. Der zurückgegebene Seitenidentifikator wird für spätere Operationen mit der Seite verwendet.

```
page1 = doc:AddPage()
```

Um eine neue Seite vor einer bestehenden Seite einzufügen, verwenden Sie den Befehl `doc:InsertPage()`. Um beispielsweise `page0` vor `page1` einzufügen, gehen Sie folgendermaßen vor:

```
page0 = doc:InsertPage(page1)
```

Nach dem Anlegen einer neuen Seite können Sie bei Bedarf einige Seitenattribute setzen. Hier stellen wir die Seitengröße auf B5 und die Ausrichtung auf Querformat ein:

```
page1:SetSize(#HPDF_PAGE_SIZE_B5, #HPDF_PAGE_LANDSCAPE)
```

Nun, da wir alles eingerichtet haben, können wir damit beginnen, der Seite Inhalte hinzuzufügen. So fügen wir beispielsweise einen "Hello World"-Text der Seite hinzu:

```
font = doc:GetFont("Times-Roman")
page0:SetFontAndSize(font, 24)
page0:BeginText()
page0:TextOut(60, 60, "Hello World!")
page0:EndText()
```

Wir können auch Grafikelemente auf die Seite zeichnen, zum Beispiel einen ausgefüllten Kreis:

```
page1:SetRGBFill(1.0, 0, 0)
```

```
page1:MoveTo(100, 100)
page1:LineTo(100, 180)
page1:Circle(100, 100, 80)
page1:Fill()
```

Wenn Sie mit dem Hinzufügen von Inhalten zu Ihren Seiten fertig sind, werden Sie wahrscheinlich das PDF-Dokument auf der Festplatte speichern wollen. Dies ist mit dem Befehl `doc:SaveToFile()` möglich. So speichern Sie unser PDF-Dokument:

```
doc:SaveToFile("test.pdf")
```

Jetzt, wo wir fertig sind, müssen wir alle Ressourcen löschen, die zum Dokumentenobjekt gehören. Dies geschieht durch Aufruf der Methode `doc:Free()` wie folgt:

```
doc:Free()
```

Beachten Sie, dass wir jetzt, da wir das Dokument und alle seine Ressourcen gelöscht haben, keine Kennungen mehr verwenden dürfen, die zu diesem Dokument gehören. In unserem Fall bedeutet dies, dass wir nicht mehr auf die folgenden Kennungen zugreifen müssen: `doc`, `page0`, `page1` und `font`. Daher ist es eine gute Idee, sie auf Null zu setzen, damit Hollywoods Speicherbereiniger sie löschen kann:

```
doc = Nil
page0 = Nil
page1 = Nil
font = Nil
```

Natürlich können Sie sie auch als lokale Variablen definieren und sie werden dann automatisch vom Speicherbereiniger gelöscht, sobald sie nicht mehr zugänglich sind.

Das war's. Herzlichen Glückwunsch, Sie haben gerade Ihr erstes PDF-Dokument mit Polybios erstellt!

6 Allgemeine Befehle

6.1 pdf.CloseDocument

BEZEICHNUNG

`pdf.CloseDocument` – schließt ein PDF-Dokument

ÜBERSICHT

```
pdf.CloseDocument(id)
```

BESCHREIBUNG

Dieser Befehl schließt ein mit `pdf.OpenDocument()` geöffnetes Dokument und gibt alle seine Ressourcen frei.

Beachten Sie, dass dieser Befehl nur für Dokumente verwendet werden darf, die mit `pdf.OpenDocument()` geöffnet wurden. Mit `pdf.CreateDocument()` erstellte Dokumente müssen mit der Methode `doc.Free()` gelöscht werden.

Beachten Sie auch, dass `pdf.CloseDocument()` nicht aufgerufen werden darf, bevor alle über `pdf.GetBrush()` enthaltenen Vektorpinsel im Dokument aus dem Speicher gelöscht wurden.

EINGABEN

`id` Identifikator des zu schließenden PDF-Dokuments

6.2 pdf.CreateDocument

BEZEICHNUNG

`pdf.CreateDocument` – erstellt ein neues PDF-Dokument

ÜBERSICHT

```
doc = pdf.CreateDocument()
```

BESCHREIBUNG

`pdf.CreateDocument()` erstellt ein neues Dokumentobjekt und gibt seine Kennung zurück. Sie können dann alle Dokumentenmethoden mit dieser Kennung verwenden. Bei einem Fehlschlag wird `Nil` zurückgegeben.

Wenn Sie mit Ihrem Dokument fertig sind, vergessen Sie nicht, `doc.Free()` darauf aufzurufen, um alle seine Ressourcen aus dem Speicher zu löschen.

EINGABEN

keine

RÜCKGABEWERTE

`doc` Dokument-Kennung

6.3 pdf.DeviceToPage

BEZEICHNUNG

pdf.DeviceToPage – konvertiert Bildschirmkoordinaten in Seitenkoordinaten (V1.2)

ÜBERSICHT

```
x, y = pdf.DeviceToPage(id, page, startx, starty, sizex, sizey, rotate,
                        devicex, devicey)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um die Bildschirmkoordinaten des durch `devicex` und `devicey` angegebenen Punkts in Seitenkoordinaten umzuwandeln.

Mit dem Argument `rotate` kann mit folgenden speziellen Werten die Seitenausrichtung angegeben werden:

- 0: Normal.
- 1: 90 Grad im Uhrzeigersinn gedreht.
- 2: Um 180 Grad gedreht.
- 3: 90 Grad gegen den Uhrzeigersinn gedreht.

Die zu verwendende Seite muss im Argument `page` als Zahl angegeben werden, welche im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegt. Die Seite muss zuvor mit `pdf.LoadPage()` und das in `id` angegebene PDF-Dokument mit `pdf.OpenDocument()` geöffnet worden sein.

Das Seitenkoordinatensystem hat seinen Ursprung in der linken unteren Ecke der Seite, wobei die X-Achse von unten links nach rechts und die Y-Achse von links unten nach oben zeigt. Beachten Sie, dass dieses Koordinatensystem beim Zoomen, Scrollen oder Drehen einer Seite geändert werden kann. Ein Punkt auf der Seite sollte jedoch immer dieselben Koordinatenwerte im Seitenkoordinatensystem haben.

Das Gerätekoordinatensystem ist geräteabhängig. Bei Bildschirmgeräten befindet sich der Ursprung in der linken oberen Ecke des Fensters.

EINGABEN

<code>id</code>	Kennung des zu verwendenden PDF-Dokuments
<code>page</code>	zu verwendende Seitenzahl (ab 1)
<code>startx</code>	linke Pixelposition des Anzeigebereichs in Gerätekoordinaten
<code>starty</code>	obere Pixelposition des Anzeigebereichs in Gerätekoordinaten
<code>sizex</code>	horizontale Größe (in Pixel) für die Anzeige der Seite
<code>sizey</code>	vertikale Größe (in Pixel) für die Anzeige der Seite
<code>rotate</code>	Seitenausrichtung (siehe oben für mögliche Werte)
<code>devicex</code>	x-Wert in zu konvertierenden Gerätekoordinaten
<code>devicey</code>	y-Wert in zu konvertierenden Gerätekoordinaten

RÜCKGABEWERTE

<code>x</code>	konvertierter x-Wert in Seitenkoordinaten
<code>y</code>	konvertierter y-Wert in Seitenkoordinaten

6.4 pdf.FindNext

BEZEICHNUNG

pdf.FindNext – sucht die nächste Instanz des Suchbegriffs (V1.1)

ÜBERSICHT

```
res = pdf.FindNext(id, page)
```

BESCHREIBUNG

Mit diesem Befehl kann ein durch `pdf.FindStart()` initiiertes Suchvorgehen fortgesetzt werden. Insbesondere findet `pdf.FindNext()` das nächste Vorkommen des an `pdf.FindStart()` übergebenen Suchbegriffs. Wenn eine andere Instanz der Suchzeichenkette gefunden werden konnte, gibt `pdf.FindNext()` `True` zurück und Sie können die Informationen darüber, wo die Zeichenkette gefunden wurde, mit `pdf.GetFindResult()` abrufen. Andernfalls wird `False` zurückgegeben.

Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments
`page` zu suchende Seitennummer (beginnend ab 1)

RÜCKGABEWERTE

`res` `True`, wenn der Suchbegriff gefunden werden konnte, sonst `False`.

6.5 pdf.FindPrev

BEZEICHNUNG

pdf.FindPrev – sucht die vorherige Instanz des Suchbegriffs (V1.1)

ÜBERSICHT

```
res = pdf.FindPrev(id, page)
```

BESCHREIBUNG

Mit diesem Befehl kann ein durch `pdf.FindStart()` initiiertes Suchvorgehen fortgesetzt werden. Insbesondere findet `pdf.FindPrev()` das vorherige Vorkommen des an `pdf.FindStart()` übergebenen Suchbegriffs. Wenn eine andere Instanz der Suchzeichenkette gefunden werden konnte, gibt `pdf.FindPrev()` `True` zurück und Sie können die Informationen darüber, wo die Zeichenkette gefunden wurde, mit `pdf.GetFindResult()` abrufen. Andernfalls wird `False` zurückgegeben.

Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments

`page` zu suchende Seitennummer (beginnend ab 1)

RÜCKGABEWERTE

`res` `True`, wenn der Suchbegriff gefunden werden konnte, sonst `False`.

6.6 pdf.FindStart

BEZEICHNUNG

`pdf.FindStart` – leitet den Suchvorgang ein (V1.1)

ÜBERSICHT

`pdf.FindStart(id, page, s$[, flags, idx])`

BESCHREIBUNG

Mit diesem Befehl kann ein neuer Suchvorgang auf der Seite gestartet werden, die durch `page` in dem durch `id` definierten Dokument angegeben ist. Sie müssen die Zeichenkette, nach der die Seite durchsucht werden soll, im Argument `s$` übergeben. Mit dem optionalen Argument `flags` können zusätzliche Optionen für den Suchvorgang konfiguriert werden. Der Parameter `flags` kann eine Kombination aus den folgenden speziellen Konstanten sein:

#PDFFIND_MATCHCASE:

Wenn dieses Flag gesetzt ist, wird der Suchvorgang unter Berücksichtigung der Groß-/Kleinschreibung durchgeführt.

#PDFFIND_MATCHWHOLEWORD:

Wenn dieses Flag gesetzt ist, wird nur dann etwas gefunden, wenn `s$` einem ganzen Wort entspricht.

Standardmäßig beginnt der Suchvorgang am Anfang der Seite. Sie können dies ändern, indem Sie im optionalen Parameter `idx` einen Zeichenindex übergeben, um die Suche an einer anderen Stelle zu starten. Beachten Sie, dass Zeichenindizes bei 0 beginnen, während die Übergabe von -1 im Parameter `idx` die Suche am Ende der Seite startet.

Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

Nachdem Sie `pdf.FindStart()` aufgerufen haben, um den Suchvorgang einzuleiten, müssen Sie dann entweder `pdf.FindNext()` oder `pdf.FindPrev()` aufrufen, um den Suchvorgang tatsächlich auszuführen.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments

`page` zu suchende Seitennummer (beginnend ab 1)

`s$` zu suchende Zeichenkette

`flags` optional: Kombination von Flags, die zusätzliche Optionen definieren (siehe oben) (Voreingestellt ist 0).

`idx` optional: Zeichenindex, bei dem die Suche gestartet werden soll (Voreingestellt ist 0)

6.7 pdf.FreePage

BEZEICHNUNG

`pdf.FreePage` – löscht eine PDF-Dokumentenseite aus dem Speicher (V1.1)

ÜBERSICHT

`pdf.FreePage(id, page)`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um eine PDF-Dokumentenseite aus dem Speicher zu löschen, die mit `pdf.LoadPage()` geladen wurde. Sie müssen die Kennung des PDF-Dokuments im Argument `id` sowie die Seitennummer, die aus dem Speicher gelöscht werden soll, im Argument `page` übergeben. Die Seitenzahl muss im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen. Das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments
`page` Seitennummer, die aus dem Speicher gelöscht werden soll (beginnend ab 1)

6.8 pdf.GetBookmarks

BEZEICHNUNG

`pdf.GetBookmarks` – ruft alle Lesezeichen in einem Dokument ab (V1.1)

ÜBERSICHT

`t = pdf.GetBookmarks(id)`

BESCHREIBUNG

Mit diesem Befehl können alle Lesezeichen in dem durch `id` angegebenen PDF-Dokument abgerufen werden. Dieses PDF-Dokument muss mit `pdf.OpenDocument()` geöffnet worden sein.

Alle Lesezeichen des Dokuments werden von `pdf.GetBookmarks()` in der Tabelle `t` zurückgegeben. Für jeden Eintrag wird die Tabelle die folgenden Felder initialisiert haben:

Title: Der Titeltext des Lesezeichens.

Action: Dieses Feld gibt an, was passieren soll, wenn auf das jeweilige Lesezeichen geklickt wird. Dieses wird auf eine der folgenden speziellen Konstanten gesetzt:

#PDFACTION_GOTO:

Springt zur Seite im aktuellen Dokument.

#PDFACTION_REMOTEGOTO:

Springt zur Seite in einem anderen Dokument.

#PDFACTION_URI:
Öffnet eine URI.

#PDFACTION_LAUNCH:
Startet ein Programm.

#PDFACTION_UNSUPPORTED:
Unbekannte Aktion.

Target: Dieses wird auf das Ziel des Lesezeichens gesetzt. Je nach **Action** kann dies auf eine Seitennummer, eine URI oder den Pfad zu einer externen Datei eingestellt werden.

Children: Wenn das Lesezeichen aufgeklappt werden kann, wird dieses Element auf eine andere Tabelle gesetzt, die die gleichen Elemente wie sein übergeordnetes Element enthält. Lesezeichen können beliebig verschachtelt werden.

EINGABEN

id Kennung des zu verwendenden PDF-Dokuments

RÜCKGABEWERTE

t Tabelle mit allen Lesezeichen des Dokuments (siehe oben)

6.9 pdf.GetBoundedText

BEZEICHNUNG

`pdf.GetBoundedText` – ermittelt den Text innerhalb des umschreibenden Rechtecks (V1.1)

ÜBERSICHT

`t$ = pdf.GetBoundedText(id, page, left, top, right, bottom)`

BESCHREIBUNG

Mit diesem Befehl kann der Text, der sich innerhalb des durch **left**, **top**, **right** und **bottom** festgelegten Begrenzungsrechtecks befindet, aus einer Seite extrahiert werden. Wenn sich kein Text innerhalb des angegebenen Begrenzungsrechtecks befindet, wird eine leere Zeichenkette zurückgegeben.

Im Argument **page** geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument **text** auf **True** geladen und das durch **id** angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

id Kennung des zu verwendenden PDF-Dokuments

page zu verwendende Seitenzahl (beginnend ab 1)

left linker Rand

top oberer Rand

`right` rechter Rand
`bottom` unterer Rand

RÜCKGABEWERTE

`t$` der Text innerhalb des umschreibenden Rechtecks

6.10 pdf.GetBrush

BEZEICHNUNG

`pdf.GetBrush` – gibt die PDF-Seite als Vektorpinsel zurück

ÜBERSICHT

```
[id, t] = pdf.GetBrush(id, page, brid[, transparent, getlinks])
```

BESCHREIBUNG

Mit diesem Befehl kann eine Seite aus dem durch `id` angegebenen PDF-Dokument in einen Vektorpinsel umgewandelt werden, der dann die ID `brid` erhält. Wenn Sie `Nil` in `brid` übergeben, wählt `pdf.GetBrush()` automatisch eine frei ID aus und gibt sie zurück.

Die zu konvertierende Seite muss im Argument `page` angegeben werden und eine Zahl im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument sein. Das von `id` angegebene PDF-Dokument muss mit `pdf.OpenDocument()` geöffnet worden sein.

Mit dem optionalen Argument `transparent` können Sie festlegen, ob der Seitenhintergrund transparent oder weiß sein soll. Wenn Sie hier `True` übergeben, erhalten Sie einen Vektorpinsel, bei dem der Seitenhintergrund durch Verwendung der Alphakanal-Transparenz vollständig transparent ist. Andernfalls wird der Seitenhintergrund weiß und Ihr Vektorpinsel wird keine Transparenz verwenden.

Beachten Sie, dass der Vektorpinsel immer noch vom PDF-Dokument abhängt, so dass es nicht erlaubt ist, `pdf.CloseDocument()` auf dem Dokument aufzurufen, während Sie den Pinsel noch benötigen.

Beachten Sie auch, dass Sie diesen Befehl nur für Seiten verwenden sollten, die noch nicht mit `pdf.LoadPage()` geladen wurden. Wenn Sie eine Seite, die mit `pdf.LoadPage()` geladen wurde, in einen Pinsel konvertieren möchten, verwenden Sie den Befehl `pdf.GetBrushFromPage()`. Siehe [Abschnitt 6.11 \[pdf.GetBrushFromPage\]](#), [Seite 28](#), für Details.

Ab Polybios 1.1 gibt es ein optionales Argument namens `getlinks`. Wenn dies auf `True` gesetzt ist, gibt `pdf.GetBrush()` eine Tabelle mit allen Links auf der Seite zurück. Die Tabelle wird als zweiter Rückgabewert zurückgegeben, wenn `getlinks` auf `True` gesetzt ist. Für jeden Eintrag wird die Tabelle die folgenden Felder initialisiert haben:

Action: Dieses Feld gibt an, was passieren soll, wenn auf den jeweiligen Link geklickt wird. Dieses wird auf eine der folgenden speziellen Konstanten gesetzt:

`#PDFACTION_GOTO:`

Zur Seite im aktuellen Dokument springen.

`#PDFACTION_REMOTEGOTO:`

Springt zur Seite in einem anderen Dokument.

#PDFACTION_URI:
Öffnet eine URI.

#PDFACTION_LAUNCH:
Startet ein Programm.

#PDFACTION_UNSUPPORTED:
Unbekannte Aktion.

Target: Dieses wird auf das Ziel des Links gesetzt. Abhängig von **Action** kann dies eingestellt werden auf eine Seitennummer, eine URI oder den Pfad zu einer externen Datei.

Left: Linker Rand des Begrenzungsrechtecks des Links.

Top: Oberer Rand des Begrenzungsrechtecks des Links.

Right: Rechter Rand des Begrenzungsrechtecks des Links.

Bottom: Unterer Rand des Begrenzungsrechtecks des Links.

EINGABEN

id Identifikator des zu verwendenden PDF-Dokuments

page zu konvertierende Seitenzahl (ab 1)

brid ID für den Vektorpinsel oder `Nil` für die automatische ID-Zuweisung

transparent
optional: **True** für einen transparenten, **False** für einen weißen Seitenhintergrund

gelinks optional: **True**, wenn Seitenlinks zurückgegeben werden sollen (siehe oben) (Standard ist **False**) (V1.1).

RÜCKGABEWERTE

id optional: Identifikator des Pinsels; wird nur zurückgegeben, wenn Sie `Nil` im dritten Argument **brid** übergeben (siehe oben).

t optional: Tabelle mit allen Seitenlinks (siehe oben) (V1.1)

6.11 pdf.GetBrushFromPage

BEZEICHNUNG

`pdf.GetBrushFromPage` – wandelt PDF-Seite in einen Vektorpinsel um (V1.1)

ÜBERSICHT

`[id] = pdf.GetBrushFromPage(id, page, brid[, transparent])`

BESCHREIBUNG

Mit diesem Befehl kann eine Seite aus dem von **id** angegebene PDF-Dokument in einen Vektorpinsel mit der Kennung **brid** umgewandelt werden. Wenn Sie `Nil` in **brid** übergeben, wählt `pdf.GetBrush()` automatisch einen freien Identifikator aus und gibt ihn zurück.

Im Argument `page` geben Sie die zu konvertierende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

Mit dem optionalen Parameter `transparent` können Sie festlegen, ob der Seitenhintergrund transparent oder weiß sein soll. Wenn Sie hier `True` übergeben, erhalten Sie einen Vektorpinsel, bei dem der Seitenhintergrund durch die Alpha-Kanal-Transparenz vollständig transparent ist. Andernfalls wird der Seitenhintergrund weiß und Ihr Vektorpinsel verwendet keine Transparenz.

Beachten Sie, dass der Vektorpinsel immer noch vom PDF-Dokument abhängt, so dass es nicht erlaubt ist `pdf.CloseDocument()` oder `pdf.FreePage()` auf dem Dokument aufzurufen, bevor der Pinsel freigegeben wird.

Wenn Sie eine PDF-Seite in einen Pinsel konvertieren möchten, ohne gezwungen zu sein, die Seite zuerst mit `pdf.LoadPage()` zu laden, verwenden Sie den Befehl `pdf.GetBrush()`. Siehe [Abschnitt 6.10 \[pdf.GetBrush\]](#), [Seite 27](#), für Details.

EINGABEN

`id` Identifikator des zu verwendenden PDF-Dokuments
`page` zu konvertierende Seitenzahl (ab 1)
`brid` ID für den Vektorpinsel oder `Nil` für die automatische ID-Zuweisung
`transparent`
 optional: `True` für einen transparenten, `False` für einen weißen Seitenhintergrund

RÜCKGABEWERTE

`id` optional: Identifikator des Pinsels; wird nur zurückgegeben, wenn Sie `Nil` im dritten Argument `brid` übergeben (siehe oben).

6.12 pdf.GetCharBox

BEZEICHNUNG

`pdf.GetCharBox` – ermittelt das umschließende Rechteck des Zeichens (V1.1)

ÜBERSICHT

`left, top, right, bottom = pdf.GetCharBox(id, page, idx)`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um den Begrenzungsrahmen des Zeichens im Index `idx` auf der durch `page` angegebenen Seite zu ermitteln. Beachten Sie, dass Zeichenindizes bei 0 beginnen, während Seitenindizes bei 1 anfangen.

Die im Argument `page` angegebene Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments

`page` zu verwendende Seitenzahl (beginnend ab 1)
`idx` Zeichenindex, dessen begrenztes Rechteck ermittelt werden soll (beginnend ab 0)

RÜCKGABEWERTE

`left` linker Rand
`top` oberer Rand
`right` rechter Rand
`bottom` unterer Rand

6.13 pdf.GetCharIndexAtPos

BEZEICHNUNG

`pdf.GetCharIndexAtPos` – ermittelt das Zeichen an der Seitenposition (V1.1)

ÜBERSICHT

`idx = pdf.GetCharIndexAtPos(id, page, x, y[, xt, yt])`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um den Index eines Zeichens an oder in der Nähe der durch `x` und `y` auf der Seite angegebenen Position zu ermitteln. Die optionalen Parameter `xt` und `yt` können verwendet werden, um einen Toleranzwert (in Punkt-Einheiten) anzugeben, der beim Ermitteln des Zeichens verwendet werden soll.

Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

`pdf.GetCharIndexAtPos()` gibt den Nullindex des Zeichens an oder in der Nähe des durch `x` und `y` angegebenen Punktes zurück. Wenn sich kein Zeichen an oder in der Nähe des Punktes befindet, ist der Rückgabewert -1 und wenn ein Fehler auftritt, wird -3 zurückgegeben.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments
`page` zu verwendende Seitenzahl (beginnend ab 1)
`x` x zu verwendende Position
`y` y zu verwendende Position
`xt` optional: x Toleranzwert (Standardwert ist 0)
`yt` optional: y Toleranzwert (Standardwert ist 0)

RÜCKGABEWERTE

`idx` Zeichenindex der angegebenen Stelle oder -1 oder -3 (siehe oben)

6.14 pdf.GetCharOrigin

BEZEICHNUNG

pdf.GetCharOrigin – ermittelt den Ursprung des Zeichens (V1.1)

ÜBERSICHT

```
x, y = pdf.GetCharOrigin(id, page, idx)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um den Ursprung des Zeichens an dem durch `idx` angegebenen Index zu ermitteln (beginnend bei 0).

Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

<code>id</code>	Kennung des zu verwendenden PDF-Dokuments
<code>page</code>	zu verwendende Seitenzahl (beginnend ab 1)
<code>idx</code>	zu verwendender Zeichenindex (beginnend mit 0)

RÜCKGABEWERTE

<code>x</code>	x Ursprungsposition
<code>y</code>	y Ursprungsposition

6.15 pdf.GetCropBox

BEZEICHNUNG

pdf.GetCropBox – gibt den Maskenrahmen des Seitenverzeichnisses zurück (V1.2)

ÜBERSICHT

```
left, top, right, bottom = pdf.GetCropBox(id, page)
```

BESCHREIBUNG

Mit diesem Befehl kann der Eintrag "CropBox" aus dem Seitenverzeichnis abgerufen werden. Die im Argument `page` angegebene Seite muss zuvor mit `pdf.LoadPage()` und das in `id` angegebene PDF-Dokument mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

<code>id</code>	Kennung des zu verwendenden PDF-Dokuments
<code>page</code>	zu verwendende Seitenzahl (beginnend ab 1)

RÜCKGABEWERTE

<code>left</code>	linker Rand
<code>top</code>	oberer Rand
<code>right</code>	rechter Rand
<code>bottom</code>	unterer Rand

6.16 pdf.GetFindResult

BEZEICHNUNG

pdf.GetFindResult – ermittelt das Ergebnis des Suchvorgangs (V1.1)

ÜBERSICHT

```
idx, len = pdf.GetFindResult(id, page)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um das Ergebnis eines Suchvorgangs zu ermitteln, nachdem pdf.FindNext() oder pdf.FindPrev() True zurückgegeben hat. In diesem Fall gibt pdf.GetFindResult() den Zeichenindex des Auftretens der Suchzeichenkette auf der Seite sowie deren Länge zurück. Zeichenindizes beginnen bei 0.

Im Argument page geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit pdf.LoadPage() mit dem Argument text auf True geladen und das durch id angegebene PDF-Dokument muss zuvor mit pdf.OpenDocument() geöffnet worden sein.

EINGABEN

id Kennung des zu verwendenden PDF-Dokuments
page zu verwendende Seitenzahl (beginnend ab 1)

RÜCKGABEWERTE

idx Startversatz des nächsten Auftretens des Suchbegriffs auf der Seite
len Länge des Suchbegriffs

6.17 pdf.GetLastError

BEZEICHNUNG

pdf.GetLastError – fragt den letzten Fehlercode ab (V1.1)

ÜBERSICHT

```
error = pdf.GetLastError()
```

BESCHREIBUNG

Wenn pdf.OpenDocument() fehlschlägt, kann pdf.GetLastError() verwendet werden, um zusätzliche Informationen zu erhalten, warum das Dokument nicht geöffnet werden konnte. Dies ist besonders nützlich, um herauszufinden, ob das Dokument nicht geöffnet werden konnte, da es passwortgeschützt ist.

pdf.GetLastError() gibt einen der folgenden Fehlercodes zurück:

```
#PDFERR_SUCCESS:
    Es ist kein Fehler aufgetreten.

#PDFERR_UNKNOWN:
    Es ist ein unbekannter Fehler aufgetreten.

#PDFERR_FILE:
    Die Datei konnte nicht gefunden werden.
```

#PDFERR_FORMAT:

Das Dateiformat konnte nicht erkannt werden.

#PDFERR_PASSWORD:

Das PDF-Dokument ist passwortgeschützt.

#PDFERR_SECURITY:

Die Sicherheitseinstellungen verbieten das Öffnen dieses Dokuments.

#PDFERR_PAGE:

Die Seitentabelle ist beschädigt.

Beachten Sie, dass Sie `pdf.GetLastError()` unmittelbar nach `pdf.OpenDocument()` aufrufen müssen, um den richtigen Ergebniscode zu erhalten.

EINGABEN

keine

RÜCKGABEWERTE

`error` letzter Fehlercode

6.18 pdf.GetMediaBox

BEZEICHNUNG

`pdf.GetMediaBox` – gibt den Medienrahmen des Seitenverzeichnisses zurück (V1.2)

ÜBERSICHT

`left, top, right, bottom = pdf.GetMediaBox(id, page)`

BESCHREIBUNG

Mit diesem Befehl kann der Eintrag "MediaBox" aus dem Seitenverzeichnis abgerufen werden. Die im Argument `page` angegebene Seite muss zuvor mit `pdf.LoadPage()` und das in `id` angegebene PDF-Dokument mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments

`page` zu verwendende Seitenzahl (beginnend ab 1)

RÜCKGABEWERTE

`left` linker Rand

`top` oberer Rand

`right` rechter Rand

`bottom` unterer Rand

6.19 pdf.GetMetaText

BEZEICHNUNG

pdf.GetMetaText – ermittelt Meta-Texte aus dem Dokument (V1.1)

ÜBERSICHT

```
t$ = pdf.GetMetaText(id, attr$)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um Metatexte aus dem durch `id` angegebene PDF-Dokument zu erhalten. Dieses PDF-Dokument muss mit `pdf.OpenDocument()` geöffnet worden sein. Das Argument `attr$` gibt an, welchen Text man erhalten soll. Dies muss eine Zeichenkette sein und kann auf die folgenden Werte gesetzt werden:

Title: Titel des Dokuments.

Author: Autor des Dokuments.

Subject: Thema des Dokuments.

Keywords:
Schlüsselwörter.

Creator Der Ersteller des Dokuments.

Producer:
Der Herausgeber des Dokuments.

CreationDate
Das Erstellungsdatum des Dokuments.

ModDate: Das Datum der letzten Änderung des Dokuments.

Beachten Sie, dass Meta-Texte nicht immer gesetzt sind. Wenn es für das angegebene Attribut keinen Metatext gibt, wird eine leere Zeichenkette zurückgegeben.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments

`attr$` Zeichenkette, die die zu ermittelnde Metadaten angibt (siehe oben für mögliche Werte).

RÜCKGABEWERTE

`t$` aus dem Dokument abgerufene Metadaten

6.20 pdf.GetObjectType

BEZEICHNUNG

pdf.GetObjectType – gibt den Objekttyp des PDF-Dokuments zurück

ÜBERSICHT

```
type = pdf.GetObjectType()
```

BESCHREIBUNG

Dieser Befehl gibt den Objekttyp zurück, der von PDF-Dokumenten verwendet wird, der mit dem Befehl `pdf.OpenDocument()` geladen wurde. Sie können diesen Objekttyp dann mit den Befehlen aus Hollywoods Objektbibliothek wie `GetAttribute()`, `SetObjectData()`, `GetObjectData()` etc. verwenden.

Insbesondere kann der Befehl `GetAttribute()` von Hollywood verwendet werden, um bestimmte Eigenschaften von dem mit `pdf.OpenDocument()` geladenen PDF-Dokumenten abzufragen. Die folgenden Attribute werden derzeit von `GetAttribute()` für PDF-Dokumente unterstützt:

#PDFATTRPAGES:

Gibt die Anzahl der Seiten im Dokument zurück.

#PDFATTRVERSION:

Liefert die PDF-Version, die dieses Dokument verwendet. Dies ist eine ganze Zahl, z.B. 14 für 1.4, 15 für 1.5, etc. (V1.1)

#PDFATTRPERMISSIONS:

Liefert eine 32-Bit Ganzzahl, die die Berechtigungskennzeichen des Dokuments beschreibt. Detaillierte Beschreibungen zu den Berechtigungen finden Sie in der PDF-Referenz. (V1.1)

EINGABEN

keine

RÜCKGABEWERTE

type interner PDF-Dokumenttyp zur Verwendung mit Hollywoods Objektbibliothek

BEISPIEL

```
pdf.OpenDocument(1, "test.pdf")
PDF_DOCUMENT = pdf.GetObjectType()
numpages = GetAttribute(PDF_DOCUMENT, 1, #PDFATTRPAGES)
```

Der obige Code öffnet `test.pdf` und fragt die Anzahl der Seiten im Dokument über `GetAttribute()` ab.

6.21 pdf.GetPageLabel**BEZEICHNUNG**

`pdf.GetPageLabel` – ruft den Seitenbeschreibungstext ab (V1.1)

ÜBERSICHT

1\$ = `pdf.GetPageLabel(id, page)`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um die Beschreibung der Seite zu erhalten, die durch das Argument `page` angegeben wird. Sie geben die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

Beachten Sie, dass Seitenbeschreibungen nicht immer gesetzt sind. Wenn es keine Beschreibung für die Seite gibt, wird eine leere Zeichenkette zurückgegeben.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments
`page` zu verwendende Seitenzahl (beginnend ab 1)

RÜCKGABEWERTE

`l$` Seitenbeschriftung

6.22 pdf.GetPageLen

BEZEICHNUNG

`pdf.GetPageLen` – ermittelt die Anzahl der Zeichen auf der Seite (V1.1)

ÜBERSICHT

```
len = pdf.GetPageLen(id, page)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um die Anzahl der Zeichen auf der Seite zu ermitteln. Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments
`page` zu verwendende Seitenzahl (beginnend ab 1)

RÜCKGABEWERTE

`len` Anzahl der Zeichen auf der Seite

6.23 pdf.GetPageLinks

BEZEICHNUNG

`pdf.GetPageLinks` – ruft alle Links auf einer PDF-Seite ab (V1.1)

ÜBERSICHT

```
t = pdf.GetPageLinks(id, page)
```

BESCHREIBUNG

Mit diesem Befehl können Sie alle Links von einer PDF-Seite abrufen. Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

Alle Links des Dokuments werden von `pdf.GetPageLinks()` in der Tabelle `t` zurückgegeben. Für jeden Eintrag wird die Tabelle die folgenden Felder initialisieren:

Action: Dieses Feld gibt an, was passieren soll, wenn auf den jeweiligen Link geklickt wird. Dieses wird auf eine der folgenden speziellen Konstanten gesetzt:

#PDFACTION_GOTO:

Zur Seite im aktuellen Dokument springen.

#PDFACTION_REMOTEGOTO:

Springt zur Seite in einem anderen Dokument.

#PDFACTION_URI:

Öffnet eine URI.

#PDFACTION_LAUNCH:

Startet ein Programm.

#PDFACTION_UNSUPPORTED:

Unbekannte Aktion.

Target: Dieses wird auf das Ziel des Links gesetzt. Je nach **Action** kann dies auf eine Seitennummer, eine URI oder den Pfad zu einer externen Datei eingestellt werden.

Left: Linker Rand des Begrenzungsrechtecks des Links.

Top: Oberer Rand des Begrenzungsrechtecks des Links.

Right: Rechter Rand des Begrenzungsrechtecks des Links.

Bottom: Unter Rand des Begrenzungsrechtecks des Links.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments

`page` zu verwendende Seitenzahl (beginnend ab 1)

RÜCKGABEWERTE

`t` Tabelle mit allen Seitenlinks (siehe oben)

6.24 pdf.GetRects

BEZEICHNUNG

`pdf.GetRects` – ermittelt das begrenzte Rechteck des Zeichenbereichs (V1.1)

ÜBERSICHT

`t = pdf.GetRects(id, page, idx, len)`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um eine Reihe von Rechtecken zu ermitteln, die den Text ab dem von `idx` angegebenen Index umschließen und sich über die Anzahl der Zeichen erstrecken. Beachten Sie, dass Zeichenindizes bei 0 beginnen. Wenn Sie -1 in `len` übergeben, erweitert `pdf.GetRects()` die Auswahl automatisch auf alle verbleibenden Zeichen.

Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

Dieser Befehl gibt eine Tabelle zurück, die eine Untertabelle pro begrenztes Rechteck enthält. Jede dieser Untertabellen hat die folgenden Felder initialisiert:

`Left:` Linker Rand.
`Top:` Oberer Rand.
`Right:` Rechter Rand.
`Bottom:` Unterer Rand.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments
`page` zu verwendende Seitenzahl (beginnend ab 1)
`idx` zu verwendender Zeichenindex (beginnend mit 0)
`len` Anzahl der zu verwendenden Zeichen oder -1 für alle übrigen Zeichen

RÜCKGABEWERTE

`t` Tabelle mit einer Reihe von begrenzten Rechtecken (siehe oben)

6.25 pdf.GetText

BEZEICHNUNG

`pdf.GetText` – ruft den Text auf der Seite ab (V1.1)

ÜBERSICHT

`t$ = pdf.GetText(id, page, idx, len)`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um den Text zu extrahieren, der bei dem von `idx` angegebenen Index beginnt und sich über die Anzahl der Zeichen einer Seite erstreckt. Beachten Sie, dass Zeichenindizes bei 0 beginnen, wenn Sie -1 in `len` übergeben, extrahiert `pdf.GetText()` automatisch alle verbleibenden Zeichen nach dem angegebenen Index.

Im Argument `page` geben Sie die zu verwendende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Die Seite muss vorher mit `pdf.LoadPage()` mit dem Argument `text` auf `True` geladen und das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

EINGABEN

`id` Kennung des zu verwendenden PDF-Dokuments
`page` zu verwendende Seitenzahl (beginnend ab 1)
`idx` zu verwendender Zeichenindex (beginnend mit 0)

`len` Anzahl der zu verwendenden Zeichen oder -1 für alle übrigen Zeichen

RÜCKGABEWERTE

`t$` Text, der extrahiert wurde

6.26 pdf.GetVersion

BEZEICHNUNG

`pdf.GetVersion` – ermittelt die libHaru-Version

ÜBERSICHT

```
ver$ = pdf.GetVersion()
```

BESCHREIBUNG

Mit diesem Befehl kann die von Polybios verwendete Version der libHaru abgefragt werden. Es wird eine Versions-Zeichenkette zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

`ver$` libHaru-Version-Zeichenkette

6.27 pdf.IsPDF

BEZEICHNUNG

`pdf.IsPDF` – überprüft, ob die Datei ein gültiges PDF-Dokument ist (V1.1)

ÜBERSICHT

```
ok = pdf.IsPDF(f$)
```

BESCHREIBUNG

Dieser Befehl prüft, ob die durch `f$` angegebene Datei im PDF-Format vorliegt und gibt `True` zurück, wenn sie es ist, ansonsten `False`.

EINGABEN

`f$` zu überprüfende Datei

RÜCKGABEWERTE

`ok` `True`, wenn die angegebene Datei ein PDF-Dokument ist, sonst `False`

6.28 pdf.LoadPage

BEZEICHNUNG

`pdf.LoadPage` – lädt eine Seite aus dem PDF-Dokument (V1.1)

ÜBERSICHT

```
pdf.LoadPage(id, page[, loadtext])
```

BESCHREIBUNG

Mit diesem Befehl kann eine Seite aus dem durch `id` angegebene PDF-Dokument geladen werden. Im Argument `page` geben Sie die zu ladende Seite als Zahl an, die im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegen muss. Das durch `id` angegebene PDF-Dokument muss zuvor mit `pdf.OpenDocument()` geöffnet worden sein.

Wenn das optionale Argument `loadtext` auf `True` gesetzt ist, lädt `pdf.LoadPage()` auch den Text der Seite. Dies ist notwendig, wenn Sie Befehle verwenden möchten, die sich mit dem Text auf einer PDF-Seite befassen, z.B. `pdf.GetText()` oder `pdf.FindStart()`.

Wenn Sie mit der Seite fertig sind, sollten Sie `pdf.FreePage()` aufrufen, um ihre Ressourcen aus dem Speicher zu löschen. Dies geschieht auch automatisch beim Aufruf von `pdf.CloseDocument()`. Siehe [Abschnitt 6.7 \[pdf.FreePage\]](#), [Seite 25](#), für Details.

EINGABEN

<code>id</code>	Kennung des zu verwendenden PDF-Dokuments
<code>page</code>	zu ladende Seitennummer (beginnend ab 1)
<code>loadtext</code>	optional: <code>True</code> , wenn der Text der Seite geladen werden soll (Standard ist <code>False</code>).

6.29 pdf.OpenDocument**BEZEICHNUNG**

`pdf.OpenDocument` – öffnet ein PDF-Dokument

ÜBERSICHT

```
[id] = pdf.OpenDocument(id, file$[, t])
```

BESCHREIBUNG

Dieser Befehl öffnet ein vorhandenes PDF-Dokument, das durch `file$` angegeben ist und weist ihm den Identifikator `id` zu. Wenn Sie `Nil` in `id` übergeben, wählt `pdf.OpenDocument()` automatisch eine freie ID und gibt sie zurück.

Mit dem optionalen Tabellenargument `t` können Sie weitere Optionen konfigurieren:

Password:

Wenn das Dokument passwortgeschützt ist, können Sie hier das Passwort zum Öffnen dieses Dokuments angeben.

Adapter: Mit diesem Tag können Sie einen oder mehrere Dateiadapter angeben, die zum Öffnen der angegebenen Datei benutzt werden sollen. Dieser Tag muss auf eine Zeichenkette gesetzt werden, der die Namen eines oder mehrerer Adapter enthält. Voreinstellung ist `default`. Weitere Informationen zu Dateiadaptern finden Sie in Ihrem Hollywood-Handbuch.

Wenn `pdf.OpenDocument()` fehlschlägt, kann `pdf.GetLastError()` verwendet werden, um zusätzliche Informationen zu erhalten, warum das Dokument nicht geöffnet werden konnte. Dies ist besonders nützlich, um herauszufinden, ob das Dokument nicht geöffnet werden konnte, da es passwortgeschützt ist. Siehe [Abschnitt 6.17 \[pdf.GetLastError\]](#), [Seite 32](#), für Details.

EINGABEN

- `id` Identifikator des zu verwendenden PDF-Dokuments oder `Nil` für die automatische ID-Zuweisung
- `file$` die zu ladende Datei
- `table` optional: Tabelle mit weiteren Optionen (siehe oben)

RÜCKGABEWERTE

- `id` optional: Identifikator des Dokuments; wird nur zurückgegeben, wenn Sie `Nil` in `id` übergeben haben (siehe oben).

6.30 pdf.PageToDevice

BEZEICHNUNG

`pdf.PageToDevice` – konvertiert Seitenkoordinaten in Bildschirmkoordinaten (V1.2)

ÜBERSICHT

```
x, y = pdf.PageToDevice(id, page, startx, starty, sizex, sizey, rotate,
                        pagex, pagey)
```

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um die Seitenkoordinaten des durch `pagex` und `pagey` angegebenen Punkts in Bildschirmkoordinaten umzuwandeln.

Mit dem Argument `rotate` kann mit folgenden speziellen Werten die Seitenausrichtung angegeben werden:

- 0: Normal.
- 1: 90 Grad im Uhrzeigersinn gedreht.
- 2: Um 180 Grad gedreht.
- 3: 90 Grad gegen den Uhrzeigersinn gedreht.

Die zu verwendende Seite muss im Argument `page` als Zahl angegeben werden, welche im Bereich von 1 bis zur Gesamtzahl der Seiten im Dokument liegt. Die Seite muss zuvor mit `pdf.LoadPage()` und das in `id` angegebene PDF-Dokument mit `pdf.OpenDocument()` geöffnet worden sein.

Das Seitenkoordinatensystem hat seinen Ursprung in der linken unteren Ecke der Seite, wobei die X-Achse von unten links nach rechts und die Y-Achse von links unten nach oben zeigt. Beachten Sie, dass dieses Koordinatensystem beim Zoomen, Scrollen oder Drehen einer Seite geändert werden kann. Ein Punkt auf der Seite sollte jedoch immer dieselben Koordinatenwerte im Seitenkoordinatensystem haben.

Das Gerätekoordinatensystem ist geräteabhängig. Bei Bildschirmgeräten befindet sich der Ursprung in der linken oberen Ecke des Fensters.

EINGABEN

- `id` Kennung des zu verwendenden PDF-Dokuments
- `page` zu verwendende Seitenzahl (ab 1)

<code>startx</code>	linke Pixelposition des Anzeigebereichs in Gerätekoordinaten
<code>starty</code>	obere Pixelposition des Anzeigebereichs in Gerätekoordinaten
<code>size<code>x</code></code>	horizontale Größe (in Pixel) für die Anzeige der Seite
<code>size<code>y</code></code>	vertikale Größe (in Pixel) für die Anzeige der Seite
<code>rotate</code>	Seitenausrichtung (siehe oben für mögliche Werte)
<code>page<code>x</code></code>	x-Wert in Seitenkoordinaten
<code>page<code>y</code></code>	y-Wert in Seitenkoordinaten

RÜCKGABEWERTE

<code>x</code>	konvertierter x-Wert in Gerätekoordinaten
<code>y</code>	konvertierter y-Wert in Gerätekoordinaten

7 Anmerkungsverfahren

7.1 annot:SetBorderStyle

BEZEICHNUNG

annot:SetBorderStyle – legt das Erscheinungsbild der Textanmerkung fest

ÜBERSICHT

```
status = annot:SetBorderStyle(subtype, width, dashon, dashoff, dashphase)
```

BESCHREIBUNG

annot:SetBorderStyle() definiert das Aussehen einer Textanmerkung. subtype muss eine der folgenden Konstanten sein:

#HPDF_BS_SOLID:

Solides Rechteck

#HPDF_BS_DASHED:

Gestricheltes Rechteck

#HPDF_BS_BEVELED:

Geprägtes Rechteck

#HPDF_BS_INSET:

Graviertes Rechteck

#HPDF_BS_UNDERLINED:

Einzelne Linie unter dem Rand der Textanmerkung

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

subtype eine der oben aufgeführten Konstanten

width die Breite des Randes einer Textanmerkung

dashon Beginn des Strichstückes

dashoff Ende des Strichstückes

dashphase

die Phase, in der das Muster beginnt

RÜCKGABEWERTE

status Statusinformation

7.2 annot:SetCMYKColor

BEZEICHNUNG

annot:SetCMYKColor – setzt die CMYK-Farbe

ÜBERSICHT

```
status = annot:SetCMYKColor(cmyk)
```

BESCHREIBUNG

Diese Methode ist zur Zeit in libHaru nicht dokumentiert. Wenden Sie sich an die libHaru-Autoren.

Der Parameter `cmyk` muss eine Tabelle mit den folgenden Feldern sein:

C	Grünblau-Anteil der Farbe (Cyan)
M	Magentarot-Anteil der Farbe (Violett/Magenta)
Y	Gelb-Anteil der Farbe (Yellow)
K	Schwarz-Anteil der Farbe (Key/Black)

Alle Werte müssen zwischen 0 und 1 liegen.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`cmyk` CMYK-Farbe

RÜCKGABEWERTE

`status` Statusinformation

7.3 annot:SetFreeTextAnnot2PointCalloutLine

BEZEICHNUNG

annot:SetFreeTextAnnot2PointCalloutLine – setzt die Freier Text-Anmerkung mit einer Zwei-Punkte-Beschreibungslinie

ÜBERSICHT

```
status = annot:SetFreeTextAnnot2PointCalloutLine(startpoint, endpoint)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Die Parameter `startpoint` und `endpoint` müssen Tabellen sein, die jeweils einen Punkt beschreiben. Jede dieser Tabellen muss also die Felder `x` und `y` enthalten.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`startpoint`
 Startpunkt

endpoint Endpunkt

RÜCKGABEWERTE

status Statusinformation

7.4 annot:SetFreeTextAnnot3PointCalloutLine

BEZEICHNUNG

annot:SetFreeTextAnnot3PointCalloutLine – setzt die Freier Text-Anmerkung mit einer Drei-Punkte-Beschreibungslinie

ÜBERSICHT

```
status = annot:SetFreeTextAnnot3PointCalloutLine(startpoint, kneepoint,
                                                    endpoint)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Die Parameter `startpoint`, `kneepoint` und `endpoint` müssen Tabellen sein, die jeweils einen Punkt beschreiben. Daher muss jede dieser Tabellen die Felder `x` und `y` enthalten.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

startpoint
 Startpunkt

kneepoint
 Kniefpunkt

endpoint Endpunkt

RÜCKGABEWERTE

status Statusinformation

7.5 annot:SetFreeTextAnnotDefaultStyle

BEZEICHNUNG

annot:SetFreeTextAnnotDefaultStyle – setzt den voreingestellten Stil der Freier Text-Anmerkung

ÜBERSICHT

```
status = annot:SetFreeTextAnnotDefaultStyle(style)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`style` voreingestellter Stil

RÜCKGABEWERTE

`status` Statusinformation

7.6 annot:SetFreeTextAnnotLineEndingStyle**BEZEICHNUNG**

`annot:SetFreeTextAnnotLineEndingStyle` – setzt den Liniendenstil der Freier Text-Anmerkung

ÜBERSICHT

```
status = annot:SetFreeTextAnnotLineEndingStyle(startstyle, endstyle)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Die Parameter `startstyle` und `endstyle` müssen eine der folgenden Konstanten sein:

```
#HPDF_LINE_ANNOT_NONE
#HPDF_LINE_ANNOT_SQUARE
#HPDF_LINE_ANNOT_CIRCLE
#HPDF_LINE_ANNOT_DIAMOND
#HPDF_LINE_ANNOT_OPENARROW
#HPDF_LINE_ANNOT_CLOSEDARROW
#HPDF_LINE_ANNOT_BUTT
#HPDF_LINE_ANNOT_ROPENARROW
#HPDF_LINE_ANNOT_RCLOSEDARROW
#HPDF_LINE_ANNOT_SLASH
```

Wir Übersetzer haben die obenstehenden Konstanten mit Erklärungen ergänzt, die so im englischen Handbuch nicht stehen. Darum sind diese Erklärungen alle ohne Gewähr.

```
#HPDF_LINE_ANNOT_NONE
```

Kein Stil am Linienende.

```
#HPDF_LINE_ANNOT_SQUARE
```

Ein Quadrat, das mit der Innenfarbe der Anmerkung gefüllt ist, falls vorhanden.

```
#HPDF_LINE_ANNOT_CIRCLE
```

Ein Kreis, der mit der Innenfarbe der Anmerkung gefüllt ist, falls vorhanden.

```
#HPDF_LINE_ANNOT_DIAMOND
```

Eine Diamantform, die mit der Innenfarbe der Anmerkung gefüllt ist, falls vorhanden.

```
#HPDF_LINE_ANNOT_OPENARROW
```

Zwei kurze Linien, die sich in einem spitzen Winkel treffen und eine offene Pfeilspitze bilden.

#HPDF_LINE_ANNOT_CLOSEDARROW
 Zwei kurze Linien, die sich in einem spitzen Winkel wie im OpenArrow-Stil treffen (siehe oben) und durch eine dritte Linie zu einer geschlossenen Pfeilspitze verbunden sind, die mit der Innenfarbe der Anmerkung gefüllt ist, falls vorhanden.

#HPDF_LINE_ANNOT_BUTT
 Eine kurze senkrecht Linie am Endpunkt der Linie.

#HPDF_LINE_ANNOT_ROPENARROW
 Zwei kurze Linien in umgekehrter Richtung wie OpenArrow.

#HPDF_LINE_ANNOT_RCLOSEDARROW
 Eine dreieckige geschlossene Pfeilspitze in umgekehrter Richtung wie CloseArrow.

#HPDF_LINE_ANNOT_SLASH
 Eine kurze Linie am Endpunkt etwa 30 Grad im Uhrzeigersinn senkrecht zur Linie.

Gibt bei Erfolg **#HPDF_OK** zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

startstyle Startstil
endstyle Endstil

RÜCKGABEWERTE

status Statusinformation

7.7 annot:SetGrayColor

BEZEICHNUNG

annot:SetGrayColor – stellt die graue Farbe ein

ÜBERSICHT

status = **annot:SetGrayColor(gray)**

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Der Parameter **gray** muss zwischen 0 und 1 liegen.

Gibt bei Erfolg **#HPDF_OK** zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

gray graue Farbe

RÜCKGABEWERTE

status Statusinformation

7.8 annot:SetLineAnnotCaption

BEZEICHNUNG

annot:SetLineAnnotCaption – setzt die Zeilenbeschriftung der Anmerkung

ÜBERSICHT

```
status = annot:SetLineAnnotCaption(show, pos, horz, vert)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Das Argument `pos` muss eine der folgenden Konstanten sein:

```
#HPDF_LINE_ANNOT_CAP_INLINE
#HPDF_LINE_ANNOT_CAP_TOP
```

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`show` Boolescher Wert, der angibt, ob die Beschriftung angezeigt werden soll (`True`) oder nicht (`False`)

`pos` Position der Beschriftung (siehe oben für mögliche Werte)

`horz` horizontaler Versatz

`vert` vertikaler Versatz

RÜCKGABEWERTE

`status` Statusinformation

7.9 annot:SetLineAnnotLeader

BEZEICHNUNG

annot:SetLineAnnotLeader – setzt die Anmerkungs-Führungslinie

ÜBERSICHT

```
status = annot:SetLineAnnotLeader(len, extlen, offsetlen)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`len` Länge

`extlen` erweiterte Länge

`offsetlen` Versatzlänge

RÜCKGABEWERTE

`status` Statusinformation

7.10 annot:SetLineAnnotPosition**BEZEICHNUNG**

`annot:SetLineAnnotPosition` – setzt die Position der Anmerkungs-Linie

ÜBERSICHT

```
status = annot:SetLineAnnotPosition(startpoint, startstyle, endpoint,
                                     endstyle)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Die Parameter `startstyle` und `endstyle` müssen eine der folgenden Konstanten sein:

```
#HPDF_LINE_ANNOT_NONE
#HPDF_LINE_ANNOT_SQUARE
#HPDF_LINE_ANNOT_CIRCLE
#HPDF_LINE_ANNOT_DIAMOND
#HPDF_LINE_ANNOT_OPENARROW
#HPDF_LINE_ANNOT_CLOSEDARROW
#HPDF_LINE_ANNOT_BUTT
#HPDF_LINE_ANNOT_ROPENARROW
#HPDF_LINE_ANNOT_RCLOSEDARROW
#HPDF_LINE_ANNOT_SLASH
```

Wir Übersetzer haben die obenstehenden Konstanten mit Erklärungen ergänzt, die so im englischen Handbuch nicht stehen. Darum sind diese Erklärungen alle ohne Gewähr. Siehe [Abschnitt 7.6 \[annot:SetFreeTextAnnotLineEndingStyle\]](#), Seite 46, für Details.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

```
startpoint      Startpunkt
startstyle      Startstil (siehe oben für mögliche Werte)
endpoint      Endpunkt
endstyle      Endstil (siehe oben für mögliche Werte)
```

RÜCKGABEWERTE

`status` Statusinformation

7.11 annot:SetLinkAnnotBorderStyle

BEZEICHNUNG

annot:SetLinkAnnotBorderStyle – setzt den Rahmenstil der Anmerkung

ÜBERSICHT

```
status = annot:SetLinkAnnotBorderStyle(width, dashon, dashoff)
```

BESCHREIBUNG

annot:SetLinkAnnotBorderStyle() definiert den Stil des Rahmens der Anmerkung.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

width die Breite des Anmerkungs-Rahmens

dashon der Strichstil

dashoff der Strichstil

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_ANNOTATION - Es wurde eine ungültige Anmerkungs-Kennung gesetzt.

#HPDF_INVALID_PARAMETER - Ein ungültiger Wert für die Breite wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung fehlgeschlagen.

7.12 annot:SetLinkAnnotHighlightMode

BEZEICHNUNG

annot:SetLinkAnnotHighlightMode – setzt die Darstellung der Hervorhebung

ÜBERSICHT

```
status = annot:SetLinkAnnotHighlightMode(mode)
```

BESCHREIBUNG

annot:SetLinkAnnotHighlightMode() definiert die Darstellung, wenn eine Maus auf einen Anmerkungs-Link klickt.

mode kann eine der folgenden Konstanten sein:

#HPDF_ANNOT_NO_HIGHLIGHT
Keine Hervorhebung.

#HPDF_ANNOT_INVERT_BOX
Invertiert den Inhalt des Anmerkungs-Bereichs.

#HPDF_ANNOT_INVERT_BORDER
Invertiert den Rahmen der Anmerkung.

#HPDF_ANNOT_DOWN_APPEARANCE
Vertieft die Anmerkung.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

mode eine der oben aufgeführten Konstanten

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_ANNOTATION - Es wurde eine ungültige Anmerkungs-Kennung gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung fehlgeschlagen.

7.13 annot:SetMarkupAnnotCloudEffect

BEZEICHNUNG

annot:SetMarkupAnnotCloudEffect – setzt den Wolkeneffekt der Anmerkungsmarkierung

ÜBERSICHT

```
status = annot:SetMarkupAnnotCloudEffect(cloudintensity)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

cloudintensity
Wolkeneffekt

RÜCKGABEWERTE

status Statusinformation

7.14 annot:SetMarkupAnnotCreationDate

BEZEICHNUNG

annot:SetMarkupAnnotCreationDate – setzt das Erstellungsdatum der Anmerkungsmarkierung

ÜBERSICHT

```
status = annot:SetMarkupAnnotCreationDate(value)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

`value` muss eine Tabelle sein, die eine Datum-/Zeitbeschreibung enthält. Die Tabelle muss folgende Felder enthalten:

`Day`: Zwischen 1 und 31 (hängt vom Monat ab).
`Month`: Zwischen 1 und 12.
`Year`: Das Jahr.
`Hour`: Zwischen 0 und 23.
`Minutes`: Zwischen 0 und 59.
`Seconds`: Zwischen 0 und 59.
`Ind`: Beziehung der lokalen Zeit zur universellen Zeit. Dies kann "", "+", "-" oder "Z" sein.
`Off_Hour`:
 Wenn `Ind` kein Leerzeichen ist, ist 0 bis 23 gültig. Andernfalls wird es ignoriert.
`Off_Minutes`:
 Wenn `Ind` kein Leerzeichen ist, ist 0 bis 59 gültig. Andernfalls wird es ignoriert.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`value` Datum-/Zeitbeschreibung

RÜCKGABEWERTE

`status` Statusinformation

7.15 `annot:SetMarkupAnnotIntent`

BEZEICHNUNG

`annot:SetMarkupAnnotIntent` – setzt das Anmerkungsmarkierungsziel

ÜBERSICHT

`status = annot:SetMarkupAnnotIntent(intent)`

BESCHREIBUNG

Diese Methode ist derzeit in `libHaru` undokumentiert. Wenden Sie sich an die `libHaru`-Autoren.

Der Parameter `intent` muss eine der folgenden Konstanten sein:

```
#HPDF_ANNOT_INTENT_FREETEXTCALLOUT
#HPDF_ANNOT_INTENT_FREETEXTTYPEWRITER
#HPDF_ANNOT_INTENT_LINEARROW
#HPDF_ANNOT_INTENT_LINEDIMENSION
#HPDF_ANNOT_INTENT_POLYGONCLOUD
#HPDF_ANNOT_INTENT_POLYLLINEDIMENSION
```

#HPDF_ANNOT_INTENT_POLYGONDIMENSION

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`intent` gewünschtes Ziel (siehe oben für mögliche Werte)

RÜCKGABEWERTE

`status` Statusinformation

7.16 annot:SetMarkupAnnotInteriorCMYKColor**BEZEICHNUNG**

`annot:SetMarkupAnnotInteriorCMYKColor` – setzt die CMYK-Innenfarbe der Anmerkungsmarkierung

ÜBERSICHT

`status = annot:SetMarkupAnnotInteriorCMYKColor(cmyk)`

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Der Parameter `cmyk` muss eine Tabelle mit folgenden initialisierten Feldern sein:

C	Grünblau-Anteil der Farbe (Cyan)
M	Magentarot-Anteil der Farbe (Violett/Magenta)
Y	Gelb-Anteil der Farbe (Yellow)
K	Schwarz-Anteil der Farbe (Key/Black)

Alle Felder müssen Werte zwischen 0 und 1 enthalten.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`cmyk` CMYK-Farbe als Tabelle

RÜCKGABEWERTE

`status` Statusinformation

7.17 annot:SetMarkupAnnotInteriorGrayColor**BEZEICHNUNG**

`annot:SetMarkupAnnotInteriorGrayColor` – setzt die graue Innenfarbe der Anmerkungsmarkierung

ÜBERSICHT

`status = annot:SetMarkupAnnotInteriorGrayColor(gray)`

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`gray` graue Farbe

RÜCKGABEWERTE

`status` Statusinformation

7.18 annot:SetMarkupAnnotInteriorRGBColor**BEZEICHNUNG**

`annot:SetMarkupAnnotInteriorRGBColor` – setzt die RGB-Innenfarbe der Anmerkungs-markierung

ÜBERSICHT

`status = annot:SetMarkupAnnotInteriorRGBColor(rgb)`

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Der Parameter `rgb` muss eine Tabelle sein, die die folgenden Felder enthält:

R Rote Farbstufe

G Grüne Farbstufe

B Blaue Farbstufe

Alle Felder müssen Werte zwischen 0 und 1 haben.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`rgb` RGB-Farbe als Tabelle

RÜCKGABEWERTE

`status` Statusinformation

7.19 annot:SetMarkupAnnotInteriorTransparent**BEZEICHNUNG**

`annot:SetMarkupAnnotInteriorTransparent` – setzt die Innentransparenz der Anmerkungs-markierung

ÜBERSICHT

`status = annot:SetMarkupAnnotInteriorTransparent()`

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

7.20 `annot:SetMarkupAnnotPopup`

BEZEICHNUNG

`annot:SetMarkupAnnotPopup` – setzt das Anmerkungsmarkierung-Popup

ÜBERSICHT

```
status = annot:SetMarkupAnnotPopup(popup)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`popup` Anmerkungsobjekt, welches als Popup benutzt werden soll

RÜCKGABEWERTE

`status` Statusinformation

7.21 `annot:SetMarkupAnnotQuadPoints`

BEZEICHNUNG

`annot:SetMarkupAnnotQuadPoints` – setzt die Viereckpunkte der Anmerkungsmarkierung

ÜBERSICHT

```
status = annot:SetMarkupAnnotQuadPoints(lb, rb, rt, lt)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Die Parameter `lb`, `rb`, `rt` und `lt` müssen Tabellen sein, die jeweils einen Punkt beschreiben. Daher muss jede dieser Tabellen die Felder `x` und `y` enthalten.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

<code>lb</code>	linker unterer Punkt
<code>rb</code>	rechter unterer Punkt
<code>rt</code>	rechter oberer Punkt
<code>lt</code>	linker oberer Punkt

RÜCKGABEWERTE

<code>status</code>	Statusinformation
---------------------	-------------------

7.22 `annot:SetMarkupAnnotRectDiff`**BEZEICHNUNG**

`annot:SetMarkupAnnotRectDiff` – setzt die Rechteckdifferenz der Anmerkungsmarkierung

ÜBERSICHT

`status = annot:SetMarkupAnnotRectDiff(rect)`

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left`, `top`, `right` und `bottom` enthält, die ein Rechteck beschreiben.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

<code>rect</code>	Tabellendefinition des Rechtecks
-------------------	----------------------------------

RÜCKGABEWERTE

<code>status</code>	Statusinformation
---------------------	-------------------

7.23 `annot:SetMarkupAnnotSubject`**BEZEICHNUNG**

`annot:SetMarkupAnnotSubject` – setzt das Thema der Anmerkungsmarkierung

ÜBERSICHT

`status = annot:SetMarkupAnnotSubject(subj)`

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

subj Thema für die Anmerkungsmarkierung

RÜCKGABEWERTE

status Statusinformation

7.24 annot:SetMarkupAnnotTitle

BEZEICHNUNG

annot:SetMarkupAnnotTitle – setzt den Titel der Anmerkungsmarkierung

ÜBERSICHT

```
status = annot:SetMarkupAnnotTitle(name)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

name Titel der Anmerkungsmarkierung

RÜCKGABEWERTE

status Statusinformation

7.25 annot:SetMarkupAnnotTransparency

BEZEICHNUNG

annot:SetMarkupAnnotTransparency – setzt die Transparenz der Anmerkungsmarkierung

ÜBERSICHT

```
status = annot:SetMarkupAnnotTransparency(value)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

value Transparenzeinstellung

RÜCKGABEWERTE

status Statusinformation

7.26 annot:SetNoColor

BEZEICHNUNG

annot:SetNoColor – stellt keine Farbe ein

ÜBERSICHT

```
status = annot:SetNoColor()
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

7.27 annot:SetPopupAnnotOpened

BEZEICHNUNG

annot:SetPopupAnnotOpened – setzt den Sichtbarkeitsstatus der Notizfenster-Anmerkung

ÜBERSICHT

```
status = annot:SetPopupAnnotOpened(open)
```

BESCHREIBUNG

annot:SetPopupAnnotOpened() definiert, ob die Notizfenster-Anmerkung zunächst geöffnet ist.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

open True bedeutet, dass die ursprünglich angezeigte Anmerkung geöffnet ist.

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_ANNOTATION - Es wurde eine ungültige Anmerkungs-Kennung gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung fehlgeschlagen.

7.28 annot:SetRGBColor

BEZEICHNUNG

annot:SetRGBColor – stellt die RGB-Farbe ein

ÜBERSICHT

```
status = annot:SetRGBColor(rgb)
```

BESCHREIBUNG

Diese Methode ist derzeit in libHaru undokumentiert. Wenden Sie sich an die libHaru-Autoren.

Der Parameter `rgb` muss eine Tabelle mit den folgenden initialisierten Feldern sein:

R	Rote Farbstufe.
G	Grüne Farbstufe.
B	Blaue Farbstufe.

Alle Werte müssen zwischen 0 und 1 liegen.

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`rgb` RGB-Farbe

RÜCKGABEWERTE

`status` Statusinformation

7.29 annot:SetTextAnnotIcon

BEZEICHNUNG

annot:SetTextAnnotIcon – setzt das Anmerkungs-Symbol

ÜBERSICHT

```
status = annot:SetTextAnnotIcon(icon)
```

BESCHREIBUNG

`annot:SetTextAnnotIcon()` definiert den Stil des Symbols der Anmerkung. `icon` kann eine der folgenden Konstanten sein:

```
#HPDF_ANNOT_ICON_COMMENT
#HPDF_ANNOT_ICON_KEY
#HPDF_ANNOT_ICON_NOTE
#HPDF_ANNOT_ICON_HELP
#HPDF_ANNOT_ICON_NEW_PARAGRAPH
#HPDF_ANNOT_ICON_PARAGRAPH
#HPDF_ANNOT_ICON_INSERT
```

Gibt bei Erfolg `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`icon` eine der oben aufgeführten Konstanten

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_ANNOTATION - Es wurde eine ungültige Anmerkungs-Kennung gesetzt.

#HPDF_ANNOT_INVALID_ICON - Ein ungültiger Symbolstil wurde angegeben.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung fehlgeschlagen.

7.30 annot:SetTextAnnotOpened**BEZEICHNUNG**

annot:SetTextAnnotOpened – setzt den Sichtbarkeitsstatus der Text-Anmerkung

ÜBERSICHT

status = annot:SetTextAnnotOpened(open)

BESCHREIBUNG

annot:SetTextAnnotOpened() definiert, ob die Text-Anmerkung zunächst geöffnet ist.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

open True bedeutet, dass die ursprünglich angezeigte Anmerkung geöffnet ist.

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_ANNOTATION - Es wurde eine ungültige Anmerkungs-Kennung gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung fehlgeschlagen.

8 Anpassungsmethoden

8.1 dest:SetFit

BEZEICHNUNG

dest:SetFit – passt die Seite innerhalb des Fensters an

ÜBERSICHT

```
status = dest:SetFit()
```

BESCHREIBUNG

dest:SetFit() legt das Erscheinungsbild der Seite fest, um die gesamte Seite innerhalb des Fensters anzuzeigen.

EINGABEN

keine

RÜCKGABEWERTE

```
status    Statusinformation
```

FEHLER

#HPDF_INVALID_DESTINATION - Eine ungültige Anpassung wurde festgelegt.

8.2 dest:SetFitB

BEZEICHNUNG

dest:SetFitB – passt den Begrenzungsrahmen der Seite innerhalb des Fensters an

ÜBERSICHT

```
status = dest:SetFitB()
```

BESCHREIBUNG

dest:SetFitB() legt das Erscheinungsbild der zu vergrößernden Seite so fest, dass sie in den Begrenzungsrahmen der Seite im Fenster passt.

EINGABEN

keine

RÜCKGABEWERTE

```
status    Statusinformation
```

FEHLER

#HPDF_INVALID_DESTINATION - Eine ungültige Anpassung wurde festgelegt.

8.3 dest:SetFitBH

BEZEICHNUNG

dest:SetFitBH – passt die Breite vom Begrenzungsrahmen an das Fenster an

ÜBERSICHT

```
status = dest:SetFitBH(top)
```

BESCHREIBUNG

`dest:SetFitBH()` definiert das Erscheinungsbild einer zu vergrößernden Seite, um sie an die Breite des Begrenzungsrahmens der Seite innerhalb des Fensters anzupassen und legt die oberste Position der Seite auf den Wert des `top`-Parameters fest.

EINGABEN

`top` die obere Koordinate der Seite

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DESTINATION` - Eine ungültige Anpassung wurde festgelegt.

`#HPDF_INVALID_PARAMETER` - Ein ungültiger Wert wurde für den `top`-Parameter festgelegt.

8.4 `dest:SetFitBV`

BEZEICHNUNG

`dest:SetFitBV` – passt die Höhe des Begrenzungsrahmens an das Fenster an

ÜBERSICHT

```
status = dest:SetFitBV(left)
```

BESCHREIBUNG

`dest:SetFitBV()` definiert das Aussehen einer zu vergrößernden Seite, um die Höhe des Begrenzungsrahmens der Seite innerhalb des Fensters anzupassen und die linke Position der Seite auf den Wert des linken Parameters einzustellen.

EINGABEN

`left` die linke Koordinate der Seite

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DESTINATION` - Eine ungültige Anpassung wurde festgelegt.

`#HPDF_INVALID_PARAMETER` - Ein ungültiger Wert wurde für den `left`-Parameter festgelegt.

8.5 `dest:SetFitH`

BEZEICHNUNG

`dest:SetFitH` – passt die Seitenbreite an das Fenster an

ÜBERSICHT

```
status = dest:SetFitH(top)
```


BESCHREIBUNG

`dest:SetFitH()` definiert das Aussehen einer zu vergrößernden Seite, um sie an die Breite der Seite innerhalb des Fensters anzupassen und legt die oberste Position der Seite auf den Wert des `top`-Parameters fest.

EINGABEN

`top` die obere Koordinate der Seite

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DESTINATION` - Eine ungültige Anpassung wurde festgelegt.

`#HPDF_INVALID_PARAMETER` - Ein ungültiger Wert wurde für den `top`-Parameter festgelegt.

8.6 `dest:SetFitR`

BEZEICHNUNG

`dest:SetFitR` – passt die Seite dem Rechteck an

ÜBERSICHT

```
status = dest:SetFitR(left, bottom, right, top)
```

BESCHREIBUNG

`dest:SetFitR()` definiert das Aussehen einer Seite, um die Seite so zu vergrößern, dass sie in ein Rechteck passt, welches durch `left`, `bottom`, `right` und `top` angegeben wird.

EINGABEN

`left` die linke Koordinate der Seite

`bottom` die untere Koordinate der Seite

`right` die rechte Koordinate der Seite

`top` die obere Koordinate der Seite

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DESTINATION` - Eine ungültige Anpassung wurde festgelegt.

`#HPDF_INVALID_PARAMETER` - Ein ungültiger Wert wurde entweder für den `left`, `bottom`, `right` und/oder `top`-Parameter festgelegt.

8.7 dest:SetFitV

BEZEICHNUNG

dest:SetFitV – passt die Seitenhöhe an das Fenster an

ÜBERSICHT

status = dest:SetFitV(left)

BESCHREIBUNG

dest:SetFitV() definiert das Erscheinungsbild einer zu vergrößernden Seite, um die Höhe der Seite innerhalb des Fensters anzupassen und setzt die linke Position der Seite auf den Wert des Parameters `left`.

EINGABEN

left die linke Koordinate der Seite

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_DESTINATION - Eine ungültige Anpassung wurde festgelegt.

#HPDF_INVALID_PARAMETER - Ein ungültiger Wert wurde für den `left`-Parameter festgelegt.

8.8 dest:SetXYZ

BEZEICHNUNG

dest:SetXYZ – definiert das Aussehen der Seite

ÜBERSICHT

status = dest:SetXYZ(left, top, zoom)

BESCHREIBUNG

dest:SetXYZ() definiert das Aussehen einer Seite mit den drei Parametern `left`, `top` und `zoom`.

EINGABEN

left die linke Koordinate der Seite

top die obere Koordinate der Seite

zoom der Seitenvergrößerungsfaktor; dieser Wert muss zwischen 0,08 (8%) und 32(3200%) liegen.

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_DESTINATION - Eine ungültige Anpassung wurde festgelegt.

#HPDF_INVALID_PARAMETER - Ein ungültiger Wert wurde entweder für den `left`, `top` oder `zoom`-Parameter festgelegt.

9 Bildmethoden

9.1 image:AddSMask

BEZEICHNUNG

image:AddSMask – fügt eine Schablonenmaske hinzu

ÜBERSICHT

```
status = image:AddSMask(smask)
```

BESCHREIBUNG

image:AddSMask() fügt ein Schablonenmaskenbild hinzu. smask muss ein Graustufenbild sein.

EINGABEN

smask Kennung eines Bildobjekts, das als Schablonenmaske verwendet werden soll.

RÜCKGABEWERTE

status Statusinformation

9.2 image:GetBitsPerComponent

BEZEICHNUNG

image:GetBitsPerComponent – ermittelt die Bits pro Farbkomponente

ÜBERSICHT

```
bpc = image:GetBitsPerComponent()
```

BESCHREIBUNG

image:GetBitsPerComponent() liefert die Anzahl der Bits, die zur Beschreibung jeder Farbkomponente verwendet werden.

EINGABEN

keine

RÜCKGABEWERTE

bpc Bits pro Farbkomponente

9.3 image:GetColorSpace()

BEZEICHNUNG

image:GetColorSpace() – ermittelt den Bildfarbraum

ÜBERSICHT

```
name = image:GetColorSpace()
```

BESCHREIBUNG

image:GetColorSpace() ruft vom Bild den Namen des Farbraums ab.

EINGABEN

keine

RÜCKGABEWERTE

name Farbraumname

9.4 image:GetHeight

BEZEICHNUNG

image:GetHeight – ermittelt die Bildhöhe

ÜBERSICHT

h = image:GetHeight()

BESCHREIBUNG

image:GetHeight() ermittelt die Höhe des Bildes eines Bildobjekts.

EINGABEN

keine

RÜCKGABEWERTE

h die Bildhöhe

9.5 image:GetSize

BEZEICHNUNG

image:GetSize – ermittelt die Bildgröße

ÜBERSICHT

w,h = image:GetSize()

BESCHREIBUNG

image:GetSize() ermittelt die Größe des Bildes eines Bildobjekts.

EINGABEN

keine

RÜCKGABEWERTE

w die Bildbreite

h die Bildhöhe

9.6 image:GetWidth

BEZEICHNUNG

image:GetWidth – ermittelt die Bildbreite

ÜBERSICHT

w = image:GetWidth()

BESCHREIBUNG

`image:GetWidth()` liefert die Breite des Bildes eines Bildobjekts.

EINGABEN

keine

RÜCKGABEWERTE

`w` die Bildbreite

9.7 image:SetColorMask**BEZEICHNUNG**

`image:SetColorMask` – stellt die transparente Farbe ein

ÜBERSICHT

```
status = image:SetColorMask(rmin, rmax, gmin, gmax, bmin, bmax)
```

BESCHREIBUNG

`image:SetColorMask()` setzt die transparente Farbe des Bildes anhand der RGB-Werte. Die Farbe innerhalb des Bereichs wird als transparente Farbe angezeigt. Das Bild muss im RGB-Farbraum sein.

EINGABEN

`rmin` die untere Grenze von Rot; sie muss zwischen 0 und 255 liegen
`rmax` die Obergrenze von Rot; sie muss zwischen 0 und 255 liegen
`gmin` die untere Grenze von Grün; sie muss zwischen 0 und 255 liegen
`gmax` die Obergrenze von Grün; sie muss zwischen 0 und 255 liegen
`bmin` die untere Grenze von Blau; sie muss zwischen 0 und 255 liegen
`bmax` die Obergrenze von Blau; sie muss zwischen 0 und 255 liegen

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_IMAGE` - Es wurde eine ungültige Bild-Kennung gesetzt.
`#HPDF_INVALID_COLOR_SPACE` - Es wurde ein Bild mit ungültigem Farbraum (nicht RGB) angegeben.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.
`#HPDF_INVALID_PARAMETER` - Ein ungültiger Wert wurde angegeben.

9.8 image:SetMaskImage**BEZEICHNUNG**

`image:SetMaskImage` – setzt das Maskenbild

ÜBERSICHT

```
status = image:SetMaskImage(maskimage)
```

BESCHREIBUNG

`image:SetMaskImage()` setzt das Maskenbild. `maskimage` muss ein 1-Bit-Graustufenbild sein.

EINGABEN

`maskimage`

Kennung eines Bildobjekts, das als Bildmaske verwendet werden soll.

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_IMAGE` - Es wurde eine ungültige Bild-Kennung gesetzt.

`#HPDF_INVALID_BIT_PER_COMPONENT` - Eine ungültige Bit-pro-Komponente.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

10 Dokumentmethoden

10.1 doc:AddPage

BEZEICHNUNG

doc:AddPage – fügt eine neue Seite zum Dokument hinzu

ÜBERSICHT

```
page = doc:AddPage()
```

BESCHREIBUNG

doc:AddPage() erstellt eine neue Seite und fügt sie nach der letzten Seite eines Dokuments hinzu.

doc:AddPage() gibt bei Erfolg die Kennung des erstellten Seitenobjekts zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

page Kennung der Seite

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

10.2 doc:AddPageLabel

BEZEICHNUNG

doc:AddPageLabel – fügt den Seitenbeschriftungsbereich hinzu

ÜBERSICHT

```
status = doc:AddPageLabel(pagenum, style, firstpage[, prefix])
```

BESCHREIBUNG

doc:AddPageLabel() fügt einen Seitenbeschriftungsbereich für das Dokument hinzu. Die Seitenbeschriftung wird in der Miniaturansicht angezeigt.

style muss eine der folgenden speziellen Konstanten sein:

#HPDF_PAGE_NUM_STYLE_DECIMAL:
Arabische Zahlen (1 2 3 4).

#HPDF_PAGE_NUM_STYLE_UPPER_ROMAN:
Römische Großbuchstabenziffern (I II III IV).

#HPDF_PAGE_NUM_STYLE_LOWER_ROMAN:
Römische Kleinbuchstabenziffern (i ii iii iv).

#HPDF_PAGE_NUM_STYLE_UPPER_LETTERS:
Großbuchstaben (A B C D).

#HPDF_PAGE_NUM_STYLE_LOWER_LETTERS:
Kleinbuchstaben (a b c d).

Wenn `doc:AddPageLabel()` erfolgreich ist, wird `#HPDF_OK` zurückgegeben. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`pagenum` die erste Seite, die diesen Beschriftungsbereich anwendet
`style` ein gültiger Nummerierungsstil (siehe oben)
`firstpage` die erste zu verwendende Seitenzahl
`prefix` optional: das Präfix für die Seitenbeschriftung

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.
`#HPDF_PAGE_NUM_STYLE_OUT_OF_RANGE` - Ein ungültiger Seitennummerierungsstil wurde angegeben.

10.3 doc:AttachFile

BEZEICHNUNG

`doc:AttachFile` – hängt eine Datei an das Dokument an

ÜBERSICHT

`file = doc:AttachFile(f$)`

BESCHREIBUNG

`doc:AttachFile()` hängt die in `f$` angegebene Datei an das Dokument an und gibt eine Kennung der eingebetteten Datei oder `Nil` bei Fehler zurück.

EINGABEN

`f$` Pfad zur Datei, die angehängt werden soll

RÜCKGABEWERTE

`file` Kennung der angehängten Datei

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

10.4 doc:CreateExtGState

BEZEICHNUNG

`doc:CreateExtGState` – erstellt ein erweitertes Grafikstatusobjekt

ÜBERSICHT

```
egs = doc:CreateExtGState()
```

BESCHREIBUNG

`doc:CreateExtGState()` erstellt ein neues erweitertes Grafikstatusobjekt.

Wenn `doc:CreateExtGState()` erfolgreich ist, wird die Kennung des erstellten erweiterten Grafikstatusobjekts zurückgegeben. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

`egs` Kennung des erweiterten Grafikstatusobjekt

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speichervuordnung ist fehlgeschlagen.

10.5 doc:CreateImageFromBrush

BEZEICHNUNG

`doc:CreateImageFromBrush` – erstellt ein neues Bild von einem Hollywood-Pinsel

ÜBERSICHT

```
img = doc:CreateImageFromBrush(id[, table])
```

BESCHREIBUNG

`doc:CreateImageFromBrush()` erstellt ein Bild aus dem in `id` angegebenen Hollywood-Pinsel. Das Bild verwendet immer den RGB-Farbraum, d.h. `#HPDF_CS_DEVICE_RGB`.

Mit dem optionalen Argument `table` können weitere Optionen konfiguriert werden:

UseJPEG: Wenn dieser Parameter auf `True` gesetzt ist, wird das Bild im JPEG-Dateiformat komprimiert. Sie können das Feld `Quality` verwenden, um die Kompressionsstufe einzustellen. Wenn `UseJPEG` auf `False` gesetzt ist, wird das Bild nicht komprimiert, aber Sie können `doc:SetCompressionMode()` verwenden, um die Kompression für Bilddaten zu aktivieren, obwohl sie nicht so gut wie JPEG ist. Standardwert ist `False`.

Quality: Hier können Sie einen Wert zwischen 0 und 100 angeben, der die Kompressionsqualität für das JPEG-Format angibt. Ein Wert von 100 bedeutet beste Qualität, 0 bedeutet schlechteste Qualität. Die Standardeinstellung ist 90, was eine ziemlich gute Qualität bedeutet.

Wenn `doc:CreateImageFromBrush()` erfolgreich ist, gibt er die Kennung eines Bildobjekts zurück. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`id` Identifikator des Pinsels, der in ein Bild konvertiert wird
`table` optional: weitere Parameter in einer Tabelle (siehe oben)

RÜCKGABEWERTE

`img` Kennung des Bildes

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speichervuordnung ist fehlgeschlagen.
`#HPDF_INVALID_COLOR_SPACE` - Ein ungültiger Farbraumwert wurde angegeben.
`#HPDF_INVALID_IMAGE` - Die Größe der Bilddaten ist ungültig.

10.6 doc:CreateImageFromMem

BEZEICHNUNG

`doc:CreateImageFromMem` – erstellt ein neues Bild aus den Speicherdaten

ÜBERSICHT

`img = doc:CreateImageFromMem(data, width, height, colorspace, bpc)`

BESCHREIBUNG

`doc:CreateImageFromMem()` erstellt ein Bild aus Rohpixeldaten im Speicher. Das Argument `data` muss ein Speicherzeiger sein, der über den Hollywood-Befehl `GetMemPointer()` ermittelt wird. Dieser Befehl lädt die Daten ohne jegliche Konvertierung, so dass er in der Regel schneller ist als die anderen Befehle. `bpc` gibt die Bitgröße jeder Farbkomponente an und kann entweder 1, 2, 4 oder 8 sein.

Das Argument `colorspace` muss die Konstante `#HPDF_CS_DEVICE_GRAY`, `#HPDF_CS_DEVICE_RGB` oder `#HPDF_CS_DEVICE_CMYK` sein. Siehe [Abschnitt 10.24 \[doc:LoadRawImage\]](#), Seite 82, für Details.

Wenn `doc:CreateImageFromMem()` erfolgreich ist, gibt er die Kennung eines Bildobjekts zurück. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`data` der Zeiger auf die Bilddaten
`width` die Breite einer Bilddatei
`height` die Höhe einer Bilddatei
`colorspace` erlaubt sind `#HPDF_CS_DEVICE_GRAY`, `#HPDF_CS_DEVICE_RGB` oder `#HPDF_CS_DEVICE_CMYK`
`bpc` die Bitgröße jeder Farbkomponente; gültige Werte sind entweder 1, 2, 4, oder 8

RÜCKGABEWERTE

`img` Kennung des Bildes

FEHLER

- #HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_INVALID_COLOR_SPACE - Ein ungültiger Farbraumwert wurde angegeben.
- #HPDF_INVALID_IMAGE - Die Größe der Bilddaten ist ungültig.

10.7 doc:CreateOutline**BEZEICHNUNG**

doc:CreateOutline – erstellt ein Umrissobjekt

ÜBERSICHT

```
otl = doc:CreateOutline(parent, title, encoder)
```

BESCHREIBUNG

doc:CreateOutline() erstellt ein neues Umrissobjekt.

Wenn doc:CreateOutline() erfolgreich ist, gibt er die Kennung des erstellten Umrissobjekts zurück. Andernfalls wird Nil zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

- parent** die Kennung des Umrissobjekts, das zum übergeordneten Element des erstellten Umrissobjekts angehängt wird; wenn Nil, wird dieses Umrissobjekt als Wurzelumrissobjekt erstellt.
- title** die Beschriftung des Umrissobjekts
- encoder** die Kennung eines Kodierungsobjekts, die auf den Titel angewendet wird; wenn Nil, wird die Kodierung des Dokuments verwendet.

RÜCKGABEWERTE

otl Kennung des Umrissobjekts

FEHLER

- #HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_INVALID_OUTLINE - Ein ungültiges übergeordnete Umrissobjekt wurde angegeben.

10.8 doc:Free**BEZEICHNUNG**

doc:Free – löscht ein Dokumentobjekt

ÜBERSICHT

```
doc:Free()
```

BESCHREIBUNG

`doc:Free()` löscht ein Dokumentobjekt und alle Ressourcen.

Beachten Sie, dass Sie nach dem Aufruf von `doc:Free()` keine zu diesem Dokument gehörende Kennungen mehr verwenden dürfen, z.B. Seiten-Kennung, Schrift-Kennung und natürlich die Dokumenten-Kennung selbst.

EINGABEN

keine

10.9 doc:GetCurrentEncoder**BEZEICHNUNG**

`doc:GetCurrentEncoder` – ermittelt den aktuellen Kodierer des Dokuments

ÜBERSICHT

```
enc = doc:GetCurrentEncoder()
```

BESCHREIBUNG

`doc:GetCurrentEncoder()` ruft die Kennung des aktuellen Kodierers des Dokumentobjekts ab. Der aktuelle Encoder wird durch Aufruf von `doc:SetCurrentEncoder()` gesetzt und dient zur Textverarbeitung, wenn eine Anwendung `doc:SetInfoAttr()` aufruft. Der Standardwert ist `Nil`.

Es wird eine Kennung eines Encoderobjekts oder `Nil` zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

`enc` Kennung des Kodierers

10.10 doc:GetCurrentPage**BEZEICHNUNG**

`doc:GetCurrentPage` – gibt das aktuelle Seitenobjekt zurück

ÜBERSICHT

```
page = doc:GetCurrentPage()
```

BESCHREIBUNG

`doc:GetCurrentPage()` gibt die Kennung des aktuellen Seitenobjekts zurück.

Wenn `doc:GetCurrentPage()` erfolgreich ist, gibt es die Kennung eines aktuellen Seitenobjekts zurück. Andernfalls wird `Nil` zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

`page` Kennung der Seite

10.11 doc:GetEncoder

BEZEICHNUNG

doc:GetEncoder – ermittelt das Kodierobjekt aus dem Namen

ÜBERSICHT

```
enc = doc:GetEncoder(encodingname)
```

BESCHREIBUNG

doc:GetEncoder() ermittelt die Kennung eines Kodierobjekts über den angegebenen Kodiernamen.

Siehe [Abschnitt 4.11 \[Kodierungen\]](#), [Seite 15](#), für eine Liste gültiger Kodierungsnamen.

Wenn doc:GetEncoder() erfolgreich ist, gibt er die Kennung eines Kodierobjekts zurück. Andernfalls wird Nil zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

encodingname
einen gültigen Kodierungsnamen (siehe oben)

RÜCKGABEWERTE

enc Kennung des Kodierers

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speichervuordnung ist fehlgeschlagen.

#HPDF_INVALID_ENCODING_NAME - Ein ungültiger Kodierungsname wurde gesetzt.

10.12 doc:GetError

BEZEICHNUNG

doc:GetError – gibt den letzten Fehlercode zurück

ÜBERSICHT

```
status = doc:GetError()
```

BESCHREIBUNG

doc:GetError() gibt den letzten Fehlercode des angegebenen Dokumentobjekts zurück.

Beachten Sie, dass einige Befehle auch einen detaillierten Fehlercode setzen. doc:GetErrorDetail() kann verwendet werden, um diesen detaillierten Fehlercode zu ermitteln.

Liefert den letzten Fehlercode des Dokumentobjekts oder #HPDF_OK, wenn kein letzter Fehler vorliegt.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

10.13 doc:GetErrorDetail

BEZEICHNUNG

doc:GetErrorDetail – ermittelt den detaillierten Fehlercode

ÜBERSICHT

```
status = doc:GetErrorDetail()
```

BESCHREIBUNG

Wenn ein Fehler auftritt, setzen einige Befehle einen detaillierten Fehlercode. doc:GetErrorDetail() gibt diesen detaillierten Fehlercode zurück.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

10.14 doc:GetFont

BEZEICHNUNG

doc:GetFont – ermittelt die Kennung des Schriftartenobjekts

ÜBERSICHT

```
font = doc:GetFont(fontname[, encodingname])
```

BESCHREIBUNG

doc:GetFont() ermittelt die Kennung eines angeforderten Schriftartenobjekts.

Siehe [Abschnitt 4.6 \[Schriftarten\]](#), [Seite 12](#), für eine Liste der gültigen Schriftartenamen.

Siehe [Abschnitt 4.11 \[Kodierungen\]](#), [Seite 15](#), für eine Liste gültiger Kodierungsnamen.

Wenn doc:GetFont() erfolgreich ist, gibt er die Kennung eines Schriftartenobjekts zurück. Andernfalls wird Nil zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

fontname ein gültiger Schriftartenname

encodingname

optional: ein gültiger Kodierungsname (voreingestellt ist die aktuelle Kodierung)

RÜCKGABEWERTE

font Kennung der Schriftart

FEHLER

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_INVALID_FONT_NAME - Ein ungültiger Schriftartenname wurde gesetzt.

#HPDF_INVALID_ENCODING_NAME - Ein ungültiger Kodierungsname wurde gesetzt.
#HPDF_UNSUPPORTED_FONT_TYPE - Es wurde ein nicht unterstützter Schriftartentyp eingestellt.

10.15 doc:GetInfoAttr

BEZEICHNUNG

doc:GetInfoAttr – holt den Text aus dem Info-Bereich

ÜBERSICHT

```
str = doc:GetInfoAttr(type)
```

BESCHREIBUNG

doc:GetInfoAttr() ruft einen Attributwert aus dem Info-Bereich ab.

Wenn doc:GetInfoAttr() erfolgreich ist, gibt er den Zeichenkettenwert des durch type angegebene Info-Bereich-Elements zurück. Wenn die Informationen nicht eingestellt wurden oder ein Fehler aufgetreten ist, wird Nil zurückgegeben.

Siehe [Abschnitt 10.32 \[doc:SetInfoAttr\]](#), Seite 88, für mögliche Typen, die an diese Methode übergeben werden können..

EINGABEN

type Info-Bereich-Element zur Abfrage

RÜCKGABEWERTE

str Text vom Info-Bereich-Element

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_INVALID_PARAMETER - Ein ungültiger Typparameter wurde gesetzt.

10.16 doc:GetPageByIndex

BEZEICHNUNG

doc:GetPageByIndex – ermittelt die Seiten-Kennung aus dem Index

ÜBERSICHT

```
page = doc:GetPageByIndex(idx)
```

BESCHREIBUNG

doc:GetPageByIndex() gibt die Seite zurück, die sich beim angegebenen Index befindet.

EINGABEN

idx Seitenindex

RÜCKGABEWERTE

page Kennung der Seite

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_INVALID_PAGE_INDEX - Der Seitenindex ist ungültig.

10.17 doc:GetPageLayout**BEZEICHNUNG**

doc:GetPageLayout – ermittelt die aktuelle Einstellung des Seitenlayouts

ÜBERSICHT

layout = doc:GetPageLayout()

BESCHREIBUNG

doc:GetPageLayout() gibt die aktuelle Einstellung für das Seitenlayout zurück.

Wenn doc:GetPageLayout() erfolgreich ist, gibt er die aktuelle Einstellung für das Seitenlayout zurück. Wenn das Seitenlayout nicht festgelegt ist, gibt er #HPDF_PAGE_LAYOUT_EOF zurück.

Siehe [Abschnitt 10.35 \[doc:SetPageLayout\]](#), Seite 90, für mögliche Seitenlayouts.

EINGABEN

keine

RÜCKGABEWERTE

layout Seitenlayoutkonstante

10.18 doc:GetPageMode**BEZEICHNUNG**

doc:GetPageMode – ermittelt den Dokumentanzeigemodus

ÜBERSICHT

mode = doc:GetPageMode()

BESCHREIBUNG

doc:GetPageMode() gibt die aktuelle Einstellung für den Seitenmodus zurück.

Siehe [Abschnitt 10.36 \[doc:SetPageMode\]](#), Seite 91, für mögliche Seitenmodi.

Wenn doc:GetPageMode() erfolgreich ist, gibt er die aktuelle Einstellung für den Seitenmodus zurück.

EINGABEN

keine

RÜCKGABEWERTE

mode aktueller Seitenmodus der Dokumentseite

10.19 doc:GetViewerPreference

BEZEICHNUNG

doc:GetViewerPreference – ermittelt die Anzeiger-Einstellungen

ÜBERSICHT

```
flags = doc:GetViewerPreference()
```

BESCHREIBUNG

doc:GetViewerPreference() ruft die Anzeiger-Einstellungen für das Dokument ab.

Siehe [Abschnitt 10.40 \[doc:SetViewerPreference\]](#), Seite 94, für eine Liste der unterstützten Einstellungen.

EINGABEN

keine

RÜCKGABEWERTE

flags Anzeiger-Einstellungen für dieses Dokument

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

10.20 doc:InsertPage

BEZEICHNUNG

doc:InsertPage – fügt eine neue Seite in das Dokument ein

ÜBERSICHT

```
page = doc:InsertPage(target)
```

BESCHREIBUNG

doc:InsertPage() erstellt eine neue Seite und fügt sie direkt vor der angegebenen Seite ein.

doc:InsertPage() gibt bei Erfolg die Kennung des neu erstellten Seitenobjekts zurück. Andernfalls wird Nil zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

page die Kennung eines Seitenobjekts, welche der Nachfolger der neuen Seite sein soll.

RÜCKGABEWERTE

page Kennung der Seite

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_INVALID_PAGE - Es wurde eine ungültige Seiten-Kennung gesetzt.

10.21 doc:LoadFont

BEZEICHNUNG

doc:LoadFont – lädt eine Schriftart mit Hollywood

ÜBERSICHT

```
font = doc:LoadFont(name[, weight, slant, embed])
```

BESCHREIBUNG

doc:LoadFont() lädt eine Schriftart mit Hollywood und registriert sie im Dokumentobjekt. Wenn das optionale Argument `embed` auf `True` gesetzt ist, werden die Glyphendaten der Schriftart eingebettet, ansonsten werden nur die Matrixdaten in die PDF-Datei aufgenommen.

Beachten Sie, dass mit dieser Methode nur TrueType-Schriftarten verwendet werden können. Sie können keine Bitmap-Schriftarten in PDF-Dokumenten verwenden.

Die optionalen Argumente `weight` und `slant` können verwendet werden, um eine Schriftstärke und eine -neigung festzulegen. Folgendes kann im Parameter `weight` übergeben werden:

```
#FONTWEIGHT_THIN
#FONTWEIGHT_EXTRALIGHT
#FONTWEIGHT_ULTRALIGHT
#FONTWEIGHT_LIGHT
#FONTWEIGHT_BOOK
#FONTWEIGHT_NORMAL (voreingestellt)
#FONTWEIGHT_REGULAR
#FONTWEIGHT_MEDIUM
#FONTWEIGHT_SEMIBOLD
#FONTWEIGHT_DEMIBOLD
#FONTWEIGHT_BOLD
#FONTWEIGHT_EXTRABOLD
#FONTWEIGHT_ULTRABOLD
#FONTWEIGHT_HEAVY
#FONTWEIGHT_BLACK
#FONTWEIGHT_EXTRABLACK
#FONTWEIGHT_ULTRABLACK
```

Die folgenden Konstanten können im Parameter `slant` übergeben werden:

```
#FONTSLANT_ROMAN (voreingestellt)
#FONTSLANT_ITALIC
#FONTSLANT_OBLIQUE
```

Wenn doc:LoadTTFont() erfolgreich ist, gibt er den Namen einer Schriftart zurück. Andernfalls wird Nil zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

<code>name</code>	Name einer Schriftart, die durch Hollywood geladen werden soll
<code>weight</code>	optional: Gewünschte Schriftstärke (Standard ist #FONTWEIGHT_NORMAL)
<code>slant</code>	optional: Gewünschte Schriftneigung (Standard ist #FONTSLANT_ROMAN)

`embed` optional: Wenn dieser Parameter auf `True` gesetzt ist, werden die Glyphendaten der Schriftart eingebettet, ansonsten werden nur die Matrixdaten in die PDF-Datei aufgenommen.

RÜCKGABEWERTE

`font` Name der Schriftart als Zeichenkette

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.
`#HPDF_FONT_EXISTS` - Die gleichnamige Schriftart ist bereits registriert.
`#HPDF_TTF_INVALID_CMAP` - Die `.ttf`-Datei konnte nicht geladen werden.
`#HPDF_TTF_INVALID_FORMAT` - Die `.ttf`-Datei konnte nicht geladen werden.
`#HPDF_TTF_MISSING_TABLE` - Die `.ttf`-Datei konnte nicht geladen werden.
`#HPDF_TTF_CANNOT_EMBEDDING_FONT` - Die Schriftart erlaubt keine Einbettung.

10.22 doc:LoadJPEGImage

BEZEICHNUNG

`doc:LoadJPEGImage` – lädt ein externes JPEG-Bild

ÜBERSICHT

```
img = doc:LoadJPEGImage(filename)
```

BESCHREIBUNG

`doc:LoadJPEGImage()` lädt eine externe JPEG-Bilddatei.

Wenn `doc:LoadJPEGImage()` erfolgreich ist, gibt er die Kennung eines Bildobjekts zurück. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`filename` Pfad zu einer JPEG-Bilddatei

RÜCKGABEWERTE

`img` Kennung des Bildes

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.
`#HPDF_UNSUPPORTED_JPEG_FORMAT` - Nicht unterstütztes JPEG-Bildformat.

10.23 doc:LoadPNGImage

BEZEICHNUNG

`doc:LoadPNGImage` – lädt ein externes PNG-Bild

ÜBERSICHT

```
img = doc:LoadPNGImage(filename[, cache])
```

BESCHREIBUNG

`doc:LoadPNGImage()` lädt eine externe PNG-Bilddatei. Mit dem optionalen Argument `cache` können Sie einstellen, ob diese Methode das gesamte PNG-Bild im Speicher zwischenspeichern soll oder nicht. Wenn Sie ein PNG-Bild mehrmals einbetten müssen, ist es schneller, dieses Argument auf `True` zu setzen.

Beachten Sie, dass beim Einbetten von PNG-Bildern in ein PDF nicht im PNG-Format, sondern als rohe, unkomprimierte Pixel eingebettet werden (obwohl Sie die Kompression für die Pixeldaten aktivieren können, indem Sie `doc:SetCompressionMode()` aufrufen). Das einzige Bildformat, das direkt in PDF-Dokumente eingebettet werden kann, ist JPEG. Verwenden Sie `doc:LoadJPEGImage()`, um ein JPEG-Bild zum Einbetten in ein PDF zu laden.

Wenn `doc:LoadPNGImage()` erfolgreich ist, gibt er die Kennung eines Bildobjekts zurück. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`filename` Pfad zu einer PNG-Bilddatei

`cache` optional: aktiviert das Caching (`True`) oder nicht (`False`) (Standard ist `False`)

RÜCKGABEWERTE

`img` Kennung des Bildes

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_UNSUPPORTED_FUNC` - Die Bibliothek ist nicht für die Verwendung von PNGLIB konfiguriert.

`#HPDF_LIBPNG_ERROR` - Fehlschlag beim Aufruf der PNGLIB-Funktion.

`#HPDF_INVALID_PNG_IMAGE` - Ungültiges PNG-Format.

10.24 doc:LoadRawImage

BEZEICHNUNG

`doc:LoadRawImage` – lädt ein Rohbild aus einer Datei

ÜBERSICHT

```
img = doc:LoadRawImage(filename, width, height, colorspace)
```

BESCHREIBUNG

`doc:LoadRawImage()` lädt ein Bild aus Rohpixeldaten, die in einer externen Datei gespeichert sind. Dieser Befehl lädt die Daten ohne Konvertierung. In der Regel ist dies schneller als die anderen Befehle. Die Pixel werden Zeile für Zeile von oben nach unten in dem durch den Parameter `colorspace` festgelegten Farbformat gespeichert, das auf eine der folgenden Konstanten eingestellt werden muss:

#HPDF_CS_DEVICE_GRAY:

8-Bit-Graustufenbild. Der Graustufen-Farbraum beschreibt jedes Pixel mit einem Byte. Für jedes Byte ist 0 die maximale Dunkelheit und 255 die maximale Helligkeit. Die Größe der Bilddaten ist `width * height` Bytes.

#HPDF_CS_DEVICE_RGB:

24 Bit RGB-Farbbild. Der 24-Bit-RGB-Farbraum beschreibt jedes Pixel mit drei Bytes (rot, grün, blau). Für jedes Byte ist 0 die maximale Dunkelheit, 255 die maximale Helligkeit. Die Größe der Bilddaten ist `width * height * 3` bytes.

#HPDF_CS_DEVICE_CMYK

32-Bit-CMYK-Farbbild. Der 32-Bit-CMYK-Farbraum beschreibt jedes Pixel mit vier Bytes (Cyan, Magenta, Gelb, Schwarz). Die Größe der Bilddaten ist `width * height * 4` bytes. Für jedes Byte ist 0 die maximale Dunkelheit, 255 die maximale Helligkeit.

Wenn `doc:LoadRawImage()` erfolgreich ist, gibt er die Kennung eines Bildobjekts zurück. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`filename` ein Pfad zu einer Bilddatei
`width` die Breite der Rohpixeldaten
`height` die Höhe der Rohpixeldaten
`colorspace`
 #HPDF_CS_DEVICE_GRAY, #HPDF_CS_DEVICE_RGB oder #HPDF_CS_DEVICE_CMYK (siehe oben)

RÜCKGABEWERTE

`img` Kennung des Bildes

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.
 #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
 #HPDF_INVALID_COLOR_SPACE - Ein ungültiger Farbraumwert wurde angegeben.
 #HPDF_INVALID_IMAGE - Die Größe der Bilddaten ist ungültig.
 #HPDF_FILE_IO_ERROR - Es können keine Daten aus der Datei gelesen werden.

10.25 doc:LoadTTFont**BEZEICHNUNG**

`doc:LoadTTFont` – lädt eine TrueType-Schriftart aus einer Datei

ÜBERSICHT

`font = doc:LoadTTFont(filename, embedding[, index])`

BESCHREIBUNG

`doc:LoadTTFont()` lädt eine TrueType-Schriftart aus einer externen Datei und registriert sie im Dokumentobjekt. Wenn das optionale Argument `index` auf einen positiven Wert gesetzt ist, lädt dieser Befehl stattdessen die TrueType-Schriftart am angegebenen Index aus einer TrueType-Sammeldatei.

Wenn `doc:LoadTTFont()` erfolgreich ist, gibt er den Namen einer Schriftart zurück. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`filename` Pfad zu einer TrueType-Schriftart (.ttf) oder TrueType-Schriftartensammlung (.ttc)

`embedding`
wenn dieser Parameter auf `True` gesetzt ist, werden die Glyphendaten der Schriftart eingebettet, ansonsten werden nur die Matrixdaten in die PDF-Datei aufgenommen.

`index` optional: Index der Schriftart, die aus der TrueType-Schriftartensammlung geladen werden soll (Voreingestellt ist -1).

RÜCKGABEWERTE

`font` Name der Schriftart als Zeichenkette

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speichervorgang ist fehlgeschlagen.

`#HPDF_FONT_EXISTS` - Die gleichnamige Schriftart ist bereits registriert.

`#HPDF_INVALID_TTC_INDEX` - Der im Index-Parameter angegebene Wert überschreitet die Anzahl der Schriftarten.

`#HPDF_INVALID_TTC_FILE` - Die .ttc-Datei konnte nicht geladen werden.

`#HPDF_TTF_INVALID_CMAP` - Die .ttf-Datei konnte nicht geladen werden.

`#HPDF_TTF_INVALID_FORMAT` - Die .ttf-Datei konnte nicht geladen werden.

`#HPDF_TTF_MISSING_TABLE` - Die .ttf-Datei konnte nicht geladen werden.

`#HPDF_TTF_CANNOT_EMBEDDING_FONT` - Die Schriftart erlaubt keine Einbettung.

10.26 doc:LoadType1Font**BEZEICHNUNG**

`doc:LoadType1Font` – lädt eine Type1-Schriftart

ÜBERSICHT

`font = doc:LoadType1Font(afmfilename, pfmfilename)`

BESCHREIBUNG

`doc:LoadType1Font()` lädt eine Type1-Schriftart aus einer externen Datei und registriert sie im Dokumentobjekt.

Wenn `doc:LoadType1Font()` erfolgreich ist, gibt er den Namen einer Schriftart zurück. Andernfalls wird `Nil` zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`afmfilename`
Pfad zur AFM-Datei

`pfmfilename`
Pfad zur PFA/PFB-Datei

RÜCKGABEWERTE

`font` Name der Schriftart als Zeichenkette

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speichervuordnung ist fehlgeschlagen.

`#HPDF_FONT_EXISTS` - Die gleichnamige Schriftart ist bereits registriert.

`#HPDF_INVALID_AFM_HEADER` - AFM-Datei kann nicht erkannt werden.

`#HPDF_INVALID_CHAR_MATRICS_DATA` - AFM-Datei kann nicht erkannt werden.

`#HPDF_INVALID_N_DATA` - AFM-Datei kann nicht erkannt werden.

`#HPDF_UNSUPPORTED_TYPE1_FONT` - PFA/PFB-Datei kann nicht erkannt werden.

10.27 doc:ResetError**BEZEICHNUNG**

`doc:ResetError` – setzt den letzten Fehlercode zurück

ÜBERSICHT

`doc:ResetError()`

BESCHREIBUNG

Sobald ein Fehlercode aufgetreten ist, können IO-Verarbeitungsbefehle nicht mehr aufgerufen werden. Im Falle der Ausführung eines Befehls, nachdem die Ursache des Fehlers behoben ist, muss eine Anwendung `doc:ResetError()` aufrufen, um den Fehlercode vor der Ausführung von Befehlen zu löschen.

EINGABEN

keine

10.28 doc:SaveToFile**BEZEICHNUNG**

`doc:SaveToFile` – speichert ein Dokument in eine Datei

ÜBERSICHT

`status = doc:SaveToFile(filename)`

BESCHREIBUNG

`doc:SaveToFile()` speichert das aktuelle Dokument in einer Datei.

Liefert bei Erfolg `#HPDF_OK`, ansonsten einen Fehlercode und der Fehler-Handler wird aufgerufen.

EINGABEN

`filename` der Name der zu speichernden Datei

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

- #HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_FILE_IO_ERROR - Bei der Verarbeitung der Datei ist ein Fehler aufgetreten.

10.29 doc:SetCompressionMode

BEZEICHNUNG

`doc:SetCompressionMode` – stellt den Dokumentenkomppressionsmodus ein

ÜBERSICHT

`status = doc:SetCompressionMode(mode)`

BESCHREIBUNG

`doc:SetCompressionMode()` setzt den Modus der Kompression. `mode` kann eine Kombination der folgenden Flags sein:

- #HPDF_COMP_NONE:
Keine Kompression. Dies kann nicht mit anderen Flags kombiniert werden.
- #HPDF_COMP_TEXT:
Komprimiert den Inhalt der Seite.
- #HPDF_COMP_IMAGE:
Komprimiert die Bildobjekte.
- #HPDF_COMP_METADATA:
Andere Daten (Schriftarten, cmaps usw.) werden komprimiert.
- #HPDF_COMP_ALL:
Alle Daten werden komprimiert. Dies entspricht dem Setzen von #HPDF_COMP_TEXT, #HPDF_COMP_IMAGE und #HPDF_COMP_METADATA zusammen.

Wenn `doc:SetCompressionMode()` erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`mode` eine Kombination der oben aufgeführten Flags

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

- #HPDF_INVALID_DOCUMENT - Es wurde eine ungültige Dokumenten-Kennung gesetzt.
- #HPDF_INVALID_COMPRESSION_MODE - Es wurde ein ungültiger Kompressionsmodus angegeben.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

10.30 doc:SetCurrentEncoder

BEZEICHNUNG

doc:SetCurrentEncoder – stellt die aktuelle Kodierung für das Dokument ein

ÜBERSICHT

```
status = doc:SetCurrentEncoder(encodingname)
```

BESCHREIBUNG

doc:SetCurrentEncoder() setzt die aktuelle Kodierung für das Dokument.

Siehe [Abschnitt 4.11 \[Kodierungen\]](#), [Seite 15](#), für eine Liste gültiger Kodierungsnamen.

Wenn doc:SetCurrentEncoder() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

encodingname
der Name einer Kodierung (siehe oben)

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_DOCUMENT - Es wurde eine ungültige Dokumenten-Kennung gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_INVALID_ENCODING_NAME - Ein ungültiger Kodierungsname wurde gesetzt.

10.31 doc:SetEncryptionMode

BEZEICHNUNG

doc:SetEncryptionMode – stellt den Dokumentenverschlüsselungsmodus ein

ÜBERSICHT

```
status = doc:SetEncryptionMode(mode[, keylen])
```

BESCHREIBUNG

doc:SetEncryptionMode() setzt den Verschlüsselungsmodus. Als Nebeneffekt erhöht es die Version von PDF auf 1.4, wenn der Modus auf #HPDF_ENCRYPT_R3 gesetzt ist.

Die folgenden Verschlüsselungsmodi werden derzeit unterstützt:

#HPDF_ENCRYPT_R2:
Verwendet den "Revision 2"-Algorithmus. keylen wird automatisch auf 5 (40 Bit) gesetzt.

#HPDF_ENCRYPT_R3:
Verwendet den "Revision 3"-Algorithmus. keylen kann 5 (40 Bit) bis 16 (128 Bit) betragen.

Wenn `doc:SetEncryptionMode()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`mode` einen der oben aufgeführten Verschlüsselungsmodi
`keylen` gibt die Bytelänge des Verschlüsselungsschlüssels an; nur erforderlich für `#HPDF_ENCRYPT_R3` (voreingestellt ist 5, z.B. 40 bits)

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Es wurde eine ungültige Dokumenten-Kennung gesetzt.
`#HPDF_INVALID_ENCRYPT_KEY_LEN` - Es wurde eine ungültige Schlüssellänge angegeben.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

10.32 doc:SetInfoAttr

BEZEICHNUNG

`doc:SetInfoAttr` – setzt den Text im Info-Bereich

ÜBERSICHT

`status = doc:SetInfoAttr(type, value)`

BESCHREIBUNG

`doc:SetInfoAttr()` setzt den Attributwert im Info-Bereich unter Verwendung der aktuellen Kodierung des Dokuments. Der Parameter `type` kann eine der folgenden Konstanten sein:

`#HPDF_INFO_AUTHOR:`
 Autor des Dokuments
`#HPDF_INFO_CREATOR:`
 Ersteller des Dokuments
`#HPDF_INFO_TITLE:`
 Titel des Dokuments
`#HPDF_INFO_SUBJECT:`
 Thema des Dokuments
`#HPDF_INFO_KEYWORDS:`
 Schlüsselwörter zur Beschreibung des Dokuments

Wenn `doc:SetInfoAttr()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`type` eine der oben aufgeführten Konstanten
`value` Text, der für die Einstellung des Attributs verwendet werden soll

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_INVALID_PARAMETER` - Ein ungültiger Parameter wurde in `type` gesetzt.

10.33 doc:SetInfoDateAttr**BEZEICHNUNG**

`doc:SetInfoDateAttr` – setzt ein Zeit-Datums-Attribut im Info-Bereich

ÜBERSICHT

`status = doc:SetInfoDateAttr(type, value)`

BESCHREIBUNG

`doc:SetInfoDateAttr()` setzt ein Zeit-Datums-Attribut im Info-Bereich. `type` muss eine der folgenden Konstanten sein:

`#HPDF_INFO_CREATION_DATE:`
Erstellungsdatum des Dokuments

`#HPDF_INFO_MOD_DATE:`
Letztes Änderungsdatum des Dokuments

`value` muss eine Tabelle mit einer Beschreibung des Datums sein. Die Tabelle muss die folgenden Felder enthalten:

`Day:` Zwischen 1 und 31 (je nach Monat).

`Month:` Zwischen 1 und 12.

`Year:` Das Jahr.

`Hour:` Zwischen 0 und 23.

`Minutes:` Zwischen 0 und 59.

`Seconds:` Zwischen 0 und 59.

`Ind:` Verhältnis der Ortszeit zur Weltzeit. Dies kann " ", "+", "-", oder "Z" sein.

`Off_Hour:`
Wenn `ind` kein Leerzeichen ist, sind 0 bis 23 gültig. Andernfalls wird `Off_Hour` ignoriert.

`Off_Minutes:`
Wenn `ind` kein Leerzeichen ist, sind 0 bis 59 gültig. Andernfalls wird `Off_Minutes` ignoriert.

EINGABEN

`type` eine der oben aufgeführten Konstanten

`value` Tabelle mit einer Beschreibung des Zeit-Datums

RÜCKGABEWERTE

`status` Statusinformation

10.34 doc:SetOpenAction**BEZEICHNUNG**

`doc:SetOpenAction` – legt die Startseite des Dokuments fest

ÜBERSICHT

`status = doc:SetOpenAction(dst)`

BESCHREIBUNG

`doc:SetOpenAction()` definiert die Seite, die beim Öffnen eines Dokuments erscheint. `dst` muss ein gültiges Zielobjekt sein, das von `page:CreateDestination()` erstellt wurde.

Wenn `doc:SetOpenAction()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`dst` gültiges Zielobjekt

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speichervuordnung ist fehlgeschlagen.

`#HPDF_INVALID_DESTINATION` - Ein ungültiges Zielobjekt wurde gesetzt.

10.35 doc:SetPageLayout**BEZEICHNUNG**

`doc:SetPageLayout` – legt fest, wie die Seiten angezeigt werden sollen

ÜBERSICHT

`status = doc:SetPageLayout(layout)`

BESCHREIBUNG

`doc:SetPageLayout()` legt fest, wie die Seiten angezeigt werden sollen. Wenn dieses Attribut nicht gesetzt ist, wird die Einstellung der Anzeige-Anwendung verwendet.

`layout` kann eine der folgenden Konstanten sein:

`#HPDF_PAGE_LAYOUT_SINGLE:`
Es wird nur eine Seite angezeigt.

`#HPDF_PAGE_LAYOUT_ONE_COLUMN:`
Zeigt die Seiten in einer Spalte an.

#HPDF_PAGE_LAYOUT_TWO_COLUMN_LEFT:

Anzeige in zwei Spalten. Die ungerade Seitenzahl wird links angezeigt.

#HPDF_PAGE_LAYOUT_TWO_COLUMN_RIGHT:

Anzeige in zwei Spalten. Die ungerade Seitenzahl wird rechts angezeigt.

Wenn `doc:SetPageLayout()` erfolgreich ist, gibt er **#HPDF_OK** zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`layout` eine der Seitenlayoutkonstanten (siehe oben)

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_PAGE_LAYOUT_OUT_OF_RANGE - Es wurde ein ungültiges Seitenlayout angegeben.

10.36 doc:SetPageMode

BEZEICHNUNG

`doc:SetPageMode` – legt fest, wie das Dokument angezeigt werden soll

ÜBERSICHT

`status = doc:SetPageMode(mode)`

BESCHREIBUNG

`doc:SetPageMode()` legt fest, wie das Dokument angezeigt werden soll.

`mode` kann eine der folgenden Konstanten sein:

#HPDF_PAGE_MODE_USE_NONE:

Zeigt das Dokument ohne Umriss und Miniaturansicht an.

#HPDF_PAGE_MODE_USE_OUTLINE:

Zeigt das Dokument mit Gliederungsfenster an.

#HPDF_PAGE_MODE_USE_THUMBS:

Zeigt das Dokument mit Miniaturansichtsfenster an.

#HPDF_PAGE_MODE_FULL_SCREEN:

Zeigt das Dokument im Vollbildmodus an.

Wenn `doc:SetPageMode()` erfolgreich ist, gibt er **#HPDF_OK** zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`mode` einen gültigen Seitenmodus (siehe oben für mögliche Optionen)

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

- #HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_PAGE_MODE_OUT_OF_RANGE - Es wurde ein ungültiger Seitenmodus angegeben.

10.37 doc:SetPagesConfiguration**BEZEICHNUNG**

doc:SetPagesConfiguration – legt die maximale Anzahl der Seiten fest

ÜBERSICHT

```
status = doc:SetPagesConfiguration(page_per_pages)
```

BESCHREIBUNG

In der Standardeinstellung hat ein Dokumentobjekt ein "Pages"-Objekt als Wurzel von Seiten. Alle "Page"-Objekte (Seite-Objekte) werden als Element des "Pages"-Objekts angelegt. Da ein "Pages"-Objekt nur 8191 Unterobjekte besitzen kann, beträgt die maximale Seitenzahl 8191 Seiten. Außerdem ist der Zustand, dass sich unter einem "Pages"-Objekt viele "Page"-Objekte befinden, nicht gut, da er zu einer Leistungseinbuße eines Anzeige-Programms führt.

Eine Anwendung kann die Einstellung eines Seitenbaums ändern, indem sie doc:SetPagesConfiguration() aufruft. Wenn der Parameter page_per_pages auf mehr als Null gesetzt ist, wird ein zweistufiger Seitenbaum erstellt. Ein Wurzelobjekt "Pages" kann 8191 "Pages"-Objekte besitzen und jedes untere "Pages"-Objekt kann page_per_pages "Page"-Objekte besitzen. Infolgedessen wird die maximale Anzahl 8191 * page_per_pages Seiten. Eine Anwendung kann doc:SetPagesConfiguration() nicht aufrufen, nachdem eine Seite zum Dokument hinzugefügt wurde.

Wenn doc:SetPagesConfiguration() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

```
page_per_pages
```

gibt die Anzahl der Seiten an, die ein "Pages"-Objekt besitzen kann.

RÜCKGABEWERTE

```
status      Statusinformation
```

FEHLER

- #HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.
- #HPDF_INVALID_DOCUMENT_STATE - Im Dokument existiert bereits ein Seitenobjekt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

10.38 doc:SetPassword**BEZEICHNUNG**

doc:SetPassword – legt das Dokumentenpasswort fest

ÜBERSICHT

```
status = doc:SetPassword(ownerpwd[, userpwd])
```

BESCHREIBUNG

`doc:SetPassword()` setzt ein Passwort für das Dokument. Wenn das Passwort festgelegt ist, werden die Dokumentinhalte verschlüsselt. Der Eigentümer kann die Berechtigung des Dokuments ändern. Beachten Sie, dass das Eigentümerkennwort nicht mit dem Benutzerkennwort übereinstimmen darf. Das Benutzerpasswort ist optional.

Wenn `doc:SetPassword()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`ownerpwd` das Passwort für den Besitzer des Dokuments

`userpwd` optional: das Passwort für den Benutzer des Dokuments

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_INVALID_PASSWORD` - Das Eigentümerpasswort ist `Nil`, eine Zeichenkette von Null Länge oder derselbe Wert wie das Benutzerpasswort.

10.39 doc:SetPermission

BEZEICHNUNG

`doc:SetPermission` – legt die Dokumentenberechtigungen fest

ÜBERSICHT

```
status = doc:SetPermission(permission)
```

BESCHREIBUNG

`doc:SetPermission()` setzt die Berechtigungsflags für das Dokument. Die Berechtigung muss aus einer Kombination der folgenden Optionen bestehen:

`#HPDF_ENABLE_READ:`

Der Benutzer kann das Dokument lesen.

`#HPDF_ENABLE_PRINT:`

Der Benutzer kann das Dokument drucken.

`#HPDF_ENABLE_EDIT_ALL:`

Der Benutzer kann den Inhalt des Dokuments bearbeiten, mit Ausnahme von Anmerkungen und Formularfeldern.

`#HPDF_ENABLE_COPY:`

Der Benutzer kann den Text und die Grafiken des Dokuments kopieren.

`#HPDF_ENABLE_EDIT:`

Der Benutzer kann die Anmerkungen und Formularfelder des Dokuments hinzufügen oder ändern.

Wenn `doc:SetPermission()` erfolgreich ist, gibt es `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

`permission`
ein oder mehrere Berechtigungsflags (siehe oben)

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

10.40 doc:SetViewerPreference

BEZEICHNUNG

`doc:SetViewerPreference` – legt die Anzeiger-Einstellungen fest

ÜBERSICHT

`status = doc:SetViewerPreference(flags)`

BESCHREIBUNG

`doc:SetViewerPreference()` setzt die Anzeiger-Einstellungen für das Dokument.
`flags` kann eine Kombination aus den folgenden Optionen sein:

`#HPDF_HIDE_TOOLBAR:`
Die Symbolleiste des Anzeigers ausblenden.

`#HPDF_HIDE_MENUBAR:`
Die Menüleiste des Anzeigers ausblenden.

`#HPDF_HIDE_WINDOW_UI`
Die Benutzeroberfläche des Anzeigers ausblenden.

`#HPDF_FIT_WINDOW:`
Dokument in das Anzeigerfenster einfügen.

`#HPDF_CENTER_WINDOW:`
Zentriert das Dokument im Anzeigerfenster.

`#HPDF_PRINT_SCALING_NONE:`
Deaktiviert die Skalierung beim Drucken.

EINGABEN

`flags` ein oder mehrere Anzeiger-Flags (siehe oben)

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

10.41 doc:UseCNSEncodings

BEZEICHNUNG

doc:UseCNSEncodings – aktiviert die vereinfachte chinesische Kodierung

ÜBERSICHT

```
status = doc:UseCNSEncodings()
```

BESCHREIBUNG

doc:UseCNSEncodings() aktiviert die vereinfachte chinesische Kodierung. Nach dem Aufruf von doc:UseCNSEncodings() kann eine Anwendung die folgenden vereinfachten chinesischen Kodierungen verwenden:

- GB-EUC-H
- GB-EUC-V
- GBK-EUC-H
- GBK-EUC-V

Wenn doc:UseCNSEncodings() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

```
status    Statusinformation
```

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_DUPLICATE_REGISTRATION - Die Kodierung des gleichen Namens wurde bereits registriert.

10.42 doc:UseCNSFonts

BEZEICHNUNG

doc:UseCNSFonts – aktiviert die vereinfachte chinesische Schriftart

ÜBERSICHT

```
status = doc:UseCNSFonts()
```

BESCHREIBUNG

doc:UseCNSFonts() aktiviert die vereinfachte chinesische Schrift. Nachdem doc:UseCNSFonts() aufgerufen wurde, kann eine Anwendung die folgenden vereinfachten chinesischen Schriftarten verwenden:

- SimSun
- SimSun,Bold
- SimSun,Italic
- SimSun,BoldItalic

- SimHei
- SimHei,Bold
- SimHei,Italic
- SimHei,BoldItalic

Wenn `doc:UseCNSFonts()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

- `#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
- `#HPDF_FAILED_TO_ALLOC_MEM` - Speichervorgang ist fehlgeschlagen.
- `#HPDF_DUPLICATE_REGISTRATION` - Die gleichnamige Schriftart ist bereits registriert.

10.43 `doc:UseCNTEncodings`

BEZEICHNUNG

`doc:UseCNTEncodings` – aktiviert die traditionelle chinesische Kodierung

ÜBERSICHT

`status = doc:UseCNTEncodings()`

BESCHREIBUNG

`doc:UseCNTEncodings()` aktiviert die traditionelle chinesische Kodierung. Nachdem `doc:UseCNTEncodings()` aufgerufen wurde, kann eine Anwendung die folgenden traditionellen chinesischen Kodierungen verwenden:

- GB-EUC-H
- GB-EUC-V
- GBK-EUC-H
- GBK-EUC-V

Wenn `doc:UseCNTEncodings()` erfolgreich ist, gibt es `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

- `#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.
- `#HPDF_FAILED_TO_ALLOC_MEM` - Speichervorgang ist fehlgeschlagen.
- `#HPDF_DUPLICATE_REGISTRATION` - Die Kodierung des gleichen Namens wurde bereits registriert.

10.44 doc:UseCNTFonts

BEZEICHNUNG

doc:UseCNTFonts – aktiviert die traditionelle chinesische Schrift

ÜBERSICHT

```
status = doc:UseCNTFonts()
```

BESCHREIBUNG

doc:UseCNTFonts() aktiviert die traditionelle chinesische Schrift. Nachdem doc:UseCNTFonts() aufgerufen wurde, kann eine Anwendung die folgenden traditionellen chinesischen Schriftarten verwenden:

- MingLiU
- MingLiU,Bold
- MingLiU,Italic
- MingLiU,BoldItalic

Wenn doc:UseCNSFonts() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

FEHLER

- #HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_DUPLICATE_REGISTRATION - Die gleichnamige Schriftart ist bereits registriert.

10.45 doc:UseJPEncodings

BEZEICHNUNG

doc:UseJPEncodings – aktiviert die japanische Kodierung

ÜBERSICHT

```
status = doc:UseJPEncodings()
```

BESCHREIBUNG

doc:UseJPEncodings() aktiviert die japanische Kodierung. Nachdem doc:UseJPEncodings() aufgerufen wurde, kann eine Anwendung die folgenden japanischen Kodierungen verwenden:

- 90ms-RKSJ-H
- 90ms-RKSJ-V
- 90msp-RKSJ-H
- EUC-H
- EUC-V

Wenn `doc:UseJPEncodings()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_DUPLICATE_REGISTRATION` - Die Kodierung des gleichen Namens wurde bereits registriert.

10.46 doc:UseJPFonts

BEZEICHNUNG

`doc:UseJPFonts` – aktiviert die japanische Schrift

ÜBERSICHT

`status = doc:UseJPFonts()`

BESCHREIBUNG

`doc:UseJPFonts()` aktiviert die japanische Schrift. Nachdem `doc:UseJPFonts()` aufgerufen wurde, kann eine Anwendung die folgenden japanischen Schriftarten verwenden:

- MS-Mincyo
- MS-Mincyo,Bold
- MS-Mincyo,Italic
- MS-Mincyo,BoldItalic
- MS-Gothic
- MS-Gothic,Bold
- MS-Gothic,Italic
- MS-Gothic,BoldItalic
- MS-PMincyo
- MS-PMincyo,Bold
- MS-PMincyo,Italic
- MS-PMincyo,BoldItalic
- MS-PGothic
- MS-PGothic,Bold
- MS-PGothic,Italic
- MS-PGothic,BoldItalic

Wenn `doc:UseJPFonts()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_DUPLICATE_REGISTRATION` - Die gleichnamige Schriftart ist bereits registriert.

10.47 doc:UseKREncodings

BEZEICHNUNG

`doc:UseKREncodings` – aktiviert die koreanische Kodierung

ÜBERSICHT

```
status = doc:UseKREncodings()
```

BESCHREIBUNG

`doc:UseKREncodings()` aktiviert die koreanische Kodierung. Nachdem `doc:UseKREncodings()` aufgerufen wurde, kann eine Anwendung die folgenden koreanischen Kodierungen verwenden:

- KSC-EUC-H
- KSC-EUC-V
- KSCms-UHC-H
- KSCms-UHC-HW-H
- KSCms-UHC-HW-V

Wenn `doc:UseKREncodings()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

`#HPDF_INVALID_DOCUMENT` - Eine ungültige Dokument-Kennung wurde festgelegt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_DUPLICATE_REGISTRATION` - Die Kodierung des gleichen Namens wurde bereits registriert.

10.48 doc:UseKRFonts

BEZEICHNUNG

doc:UseKRFonts – aktiviert die koreanische Schrift

ÜBERSICHT

```
status = doc:UseKRFonts()
```

BESCHREIBUNG

doc:UseKRFonts() aktiviert die koreanische Schrift. Nachdem doc:UseKRFonts() aufgerufen wurde, kann eine Anwendung die folgenden koreanischen Schriftarten verwenden:

- DotumChe
- DotumChe,Bold
- DotumChe,Italic
- DotumChe,BoldItalic
- Dotum
- Dotum,Bold
- Dotum,Italic
- Dotum,BoldItalic
- BatangChe
- BatangChe,Bold
- BatangChe,Italic
- BatangChe,BoldItalic
- Batang
- Batang,Bold
- Batang,Italic
- Batang,BoldItalic

Wenn doc:UseKRFonts() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_DUPLICATE_REGISTRATION - Die gleichnamige Schriftart ist bereits registriert.

10.49 doc:UseUTFEncodings

BEZEICHNUNG

doc:UseUTFEncodings – aktiviert die UTF-8 Kodierung

ÜBERSICHT

```
status = doc:UseUTFEncodings()
```

BESCHREIBUNG

doc:UseUTFEncodings() aktiviert die UTF-8 Kodierung. Nachdem der Befehl doc:UseUTFEncodings() aufgerufen wurde, kann eine Anwendung UTF-8-kodierten Unicode-Text enthalten (nur bis zu 3 Byte UTF-8-Sequenzen). Eine Anwendung kann die folgenden Unicode-Kodierungen verwenden (jedoch nur mit TrueType-Schriftarten):

- UTF-8

Wenn doc:UseUTFEncodings() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

```
status    Statusinformation
```

FEHLER

#HPDF_INVALID_DOCUMENT - Eine ungültige Dokument-Kennung wurde festgelegt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_DUPLICATE_REGISTRATION - Die Kodierung des gleichen Namens wurde bereits registriert.

11 Grafikzustandsmethoden

11.1 extgs:SetAlphaFill

BEZEICHNUNG

extgs:SetAlphaFill – stellt die Alphafülltransparenz ein

ÜBERSICHT

status = extgs:SetAlphaFill(value)

BESCHREIBUNG

extgs:SetAlphaFill() definiert die Transparenz für das Ausfüllen.

Wenn extgs:SetAlphaFill() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

value der Alpha-Wert zum Ausfüllen; er muss zwischen 0 und 1 liegen

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_OBJECT - Eine ungültige Grafikzustand-Kennung wurde gesetzt.

#HPDF_EXT_GSTATE_READ_ONLY - Das Grafikzustandsobjekt kann nur gelesen werden.

#HPDF_EXT_GSTATE_OUT_OF_RANGE - Im Parameter value wurde ein ungültiger Wert gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

11.2 extgs:SetAlphaStroke

BEZEICHNUNG

extgs:SetAlphaStroke – setzt die Zeichnungstransparenz

ÜBERSICHT

status = extgs:SetAlphaStroke(value)

BESCHREIBUNG

extgs:SetAlphaStroke() definiert die Transparenz für das Zeichnen.

Wenn extgs:SetAlphaStroke() erfolgreich ist, gibt er #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

value der Alpha-Wert für das Zeichnen; er muss zwischen 0 und 1 liegen

RÜCKGABEWERTE

status Statusinformation

FEHLER

- #HPDF_INVALID_OBJECT - Eine ungültige Grafikzustand-Kennung wurde gesetzt.
- #HPDF_EXT_GSTATE_READ_ONLY - Das Grafikzustandsobjekt kann nur gelesen werden.
- #HPDF_EXT_GSTATE_OUT_OF_RANGE - Im Parameter `value` wurde ein ungültiger Wert gesetzt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

11.3 extgs:SetBlendMode**BEZEICHNUNG**

`extgs:SetBlendMode` – stellt den Mischmodus ein

ÜBERSICHT

`status = extgs:SetBlendMode(bmode)`

BESCHREIBUNG

`extgs:SetBlendMode()` setzt die Methode des Mischmodus.

Der Parameter `bmode` muss eine der folgenden Konstanten sein:

```
#HPDF_BM_NORMAL
#HPDF_BM_MULTIPLY
#HPDF_BM_SCREEN
#HPDF_BM_OVERLAY
#HPDF_BM_DARKEN
#HPDF_BM_LIGHTEN
#HPDF_BM_COLOR_DODGE
#HPDF_BM_COLOR_BUM
#HPDF_BM_HARD_LIGHT
#HPDF_BM_SOFT_LIGHT
#HPDF_BM_DIFFERENCE
#HPDF_BM_EXCLUSHON
```

Wenn `extgs:SetBlendMode()` erfolgreich ist, gibt er `#HPDF_OK` zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler aufgerufen.

EINGABEN

`bmode` gewünschter Mischmodus (siehe oben für mögliche Werte)

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

- #HPDF_INVALID_OBJECT - Eine ungültige Grafikzustand-Kennung wurde gesetzt.
- #HPDF_EXT_GSTATE_READ_ONLY - Das Grafikzustandsobjekt kann nur gelesen werden.
- #HPDF_EXT_GSTATE_OUT_OF_RANGE - Im Parameter `value` wurde ein ungültiger Wert gesetzt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

12 Kodiermethoden

12.1 encoder:GetByteType

BEZEICHNUNG

encoder:GetByteType – ermittelt den Byte-Typ im Text

ÜBERSICHT

```
t = encoder:GetByteType(text, index)
```

BESCHREIBUNG

encoder:GetByteType() gibt den Typ des Bytes im Text an der angegebenen Position index zurück.

EINGABEN

text Textzeichenkette
index Index innerhalb der Textzeichenkette

RÜCKGABEWERTE

t byte type

12.2 encoder:GetType

BEZEICHNUNG

encoder:GetType – ermittelt den Typ des Kodierungsobjekts

ÜBERSICHT

```
t = encoder:GetType()
```

BESCHREIBUNG

encoder:GetType() ruft den Typ eines Kodierungsobjekts ab.

EINGABEN

keine

RÜCKGABEWERTE

t Kodiertyp

12.3 encoder:GetUnicode

BEZEICHNUNG

encoder:GetUnicode – konvertiert Zeichen in Unicode

ÜBERSICHT

```
unicode = encoder:GetUnicode(code)
```

BESCHREIBUNG

encoder:GetUnicode() konvertiert einen bestimmten Zeichencode in Unicode.

EINGABEN

`code` ein Zeichencode zur Konvertierung

RÜCKGABEWERTE

`ucode` Zeichencode in Unicode

12.4 `encoder:GetWritingMode`

BEZEICHNUNG

`encoder:GetWritingMode` – ermittelt den Schreibmodus des Kodierungsobjekts

ÜBERSICHT

```
mode = encoder:GetWritingMode()
```

BESCHREIBUNG

`encoder:GetWritingMode()` gibt den Schreibmodus für das Kodierungsobjekt zurück.

EINGABEN

keine

RÜCKGABEWERTE

`mode` Schreibmodus

13 Schriftartmethoden

13.1 font:GetAscent

BEZEICHNUNG

font:GetAscent – ermittelt den vertikalen Aufstieg der Schriftart

ÜBERSICHT

```
asc = font:GetAscent()
```

BESCHREIBUNG

font:GetAscent() ruft den vertikalen Anstieg der Schriftart ab.

Gibt die vertikale Steigung der Schrift bei Erfolg zurück. Andernfalls wird 0 zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

asc vertikaler Aufstieg

13.2 font:GetBBox

BEZEICHNUNG

font:GetBBox – ermittelt den Schriftarten-Begrenzungsrahmen

ÜBERSICHT

```
bbox = font:GetBBox()
```

BESCHREIBUNG

font:GetBBox() ruft den Begrenzungsrahmen der Schriftart ab. Dieser Befehl gibt eine Tabelle zurück, in der die Felder **left** (links), **top** (oben), **right** (rechts) und **bottom** (unten) initialisiert sind. Bei Erfolg werden die Felder auf den Begrenzungsrahmen der Schriftart gesetzt, ansonsten sind alle Felder 0.

EINGABEN

keine

RÜCKGABEWERTE

bbox Begrenzungsrahmen der Schriftart

13.3 font:GetCapHeight

BEZEICHNUNG

font:GetCapHeight – ermittelt den Großbuchstaben-Grundlinienabstand

ÜBERSICHT

```
ch = font:GetCapHeight()
```

BESCHREIBUNG

`font:GetCapHeight()` ermittelt bei Großbuchstaben den Abstand von der Grundlinie.
Gibt bei Erfolg die Versalhöhe zurück. Andernfalls wird 0 zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

ch Versalhöhe

13.4 font:GetDescent**BEZEICHNUNG**

`font:GetDescent` – ermittelt den vertikalen Abstieg der Schriftart

ÜBERSICHT

`desc = font:GetDescent()`

BESCHREIBUNG

`font:GetDescent()` ruft den vertikalen Abstieg der Schriftart ab.

Gibt bei Erfolg den senkrechten Abstieg der Schrift zurück. Andernfalls wird 0 zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

desc vertikaler Abstieg

13.5 font:GetEncodingName**BEZEICHNUNG**

`font:GetEncodingName` – ermittelt den Kodierungsnamen der Schriftart

ÜBERSICHT

`name = font:GetEncodingName()`

BESCHREIBUNG

`font:GetEncodingName()` ruft den Kodierungsnamen der Schriftart ab.

Liefert bei Erfolg den Namen der Schriftartkodierung. Andernfalls wird Nil zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

name Name der Schriftartkodierung

13.6 font:GetFontName

BEZEICHNUNG

font:GetFontName – ermittelt den Schriftartennamen

ÜBERSICHT

```
name = font:GetFontName()
```

BESCHREIBUNG

font:GetFontName() ruft den Namen der Schriftart ab.

Gibt bei Erfolg den Schriftartennamen zurück. Andernfalls wird Nil zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

name Schriftartname

13.7 font:GetUnicodeWidth

BEZEICHNUNG

font:GetUnicodeWidth – ermittelt die Unicode-Zeichenbreite

ÜBERSICHT

```
w = font:GetUnicodeWidth(code)
```

BESCHREIBUNG

font:GetUnicodeWidth() ermittelt die Breite eines Unicode-Zeichens in einer bestimmten Schriftart. Die tatsächliche Breite des Zeichens auf der Seite kann wie folgt berechnet werden:

```
char_width = font:GetUnicodeWidth(font, UNICODE)
actual_width = char_width * FONT_SIZE / 1000
```

Gibt bei Erfolg die Zeichenbreite zurück. Andernfalls wird Nil zurückgegeben.

EINGABEN

code ein Unicode-Zeichen

RÜCKGABEWERTE

w Unicode-Zeichenbreite

13.8 font:GetXHeight

BEZEICHNUNG

font:GetXHeight – ermittelt den Abstand von der Grundlinie bei Kleinbuchstaben

ÜBERSICHT

```
xh = font:GetXHeight()
```

BESCHREIBUNG

`font:GetXHeight()` ermittelt bei Kleinbuchstaben den Abstand zur Grundlinie.
Liefert bei Erfolg den Abstand von der Grund- zur x-Linie der Schriftart (Mittellänge/x-Höhe). Andernfalls wird 0 zurückgegeben.

EINGABEN

keine

RÜCKGABEWERTE

`xh` x-Höhe/Mittellänge

13.9 font:MeasureText**BEZEICHNUNG**

`font:MeasureText` – berechnet die Länge von Textbyte

ÜBERSICHT

```
bl, rw = font:MeasureText(text, len, width, fontsize, charspace,
                          wordspace, wordwrap)
```

BESCHREIBUNG

`font:MeasureText()` berechnet die Bytelänge, die innerhalb der angegebenen Breite enthalten ist.

Der Parameter `wordwrap` legt fest, wie Wörter umgebrochen werden sollen: Angenommen es gibt drei Wörter: "ABCDE", "FGH" und "IJKL". Angenommen die Teilzeichenkette wird so lange verwendet, bis "J" innerhalb der Breite (12 Byte) enthalten sein kann. Wenn `wordwrap` `False` ist, gibt der Befehl 12 zurück. Wenn der Parameter `wordwrap` `True` ist, gibt er 10 zurück (das Ende des vorherigen Wortes).

Bei Erfolg wird die Bytelänge zurückgegeben, die innerhalb der angegebenen Breite berücksichtigt wird. Andernfalls wird 0 zurückgegeben.

EINGABEN

`text` der Text, der für die Berechnung verwendet werden soll
`len` die Länge des Textes
`width` die Breite des Bereichs, in dem der Text platziert werden soll
`fontsize` die Größe der Schriftart
`charspace` der Zeichenabstand
`wordspace` der Wortabstand
`wordwrap` Boolescher Wert, ob ein Zeilenumbruch aktiviert werden soll oder nicht.

RÜCKGABEWERTE

`bl` Bytelänge
`rw` tatsächliche Breite des Textes

13.10 font:TextWidth

BEZEICHNUNG

font:TextWidth – ermittelt die Textbreite des Textes

ÜBERSICHT

```
t = font:TextWidth(text, len)
```

BESCHREIBUNG

font:TextWidth() ruft die Gesamtbreite des Textes, die Anzahl der Zeichen und die Anzahl der Wörter ab.

Diese Methode gibt eine Tabelle zurück, in der folgende Felder initialisiert sind:

NumChars:

Die Anzahl der Zeichen.

NumWords:

Die Anzahl der Wörter (veraltet). Verwenden Sie stattdessen NumSpace (siehe unten).

Width: Die Gesamtbreite des Textes.

NumSpace:

Die Anzahl der Wörter.

Im Falle eines Fehlers werden alle Tabellenelemente auf 0 gesetzt.

EINGABEN

text der Text, dessen Breite ermittelt werden soll

len die Bytelänge des Textes

RÜCKGABEWERTE

t Tabelle mit den Berechnungsergebnissen (siehe oben)

14 Seitenmethoden

14.1 page:Arc

BEZEICHNUNG

page:Arc – fügt einen Bogen an den Pfad an

ÜBERSICHT

status = page:Arc(x, y, radius, ang1, ang2)

BESCHREIBUNG

page:Arc() hängt einen Kreisbogen an den aktuellen Pfad an. Die Winkel werden in Grad angegeben, wobei 0 Grad vertikal von der Position (x, y) nach oben gerichtet ist.

EINGABEN

x	x-Mittelpunkt des Bogens
y	y-Mittelpunkt des Bogens
radius	der Radius des Bogens
ang1	Anfangswinkel des Bogens
ang2	Endwinkel des Bogens; er muss größer als ang1 sein

RÜCKGABEWERTE

status Statusinformation

14.2 page:BeginText

BEZEICHNUNG

page:BeginText – beginnt ein Textobjekt

ÜBERSICHT

status = page:BeginText()

BESCHREIBUNG

page:BeginText() beginnt ein Textobjekt und setzt die Textposition auf (0, 0).

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.3 page:Circle

BEZEICHNUNG

page:Circle – fügt einen Kreis an den Pfad an

ÜBERSICHT

status = page:Circle(x, y, radius)

BESCHREIBUNG

page:Circle() hängt einen Kreis an den aktuellen Pfad an.

EINGABEN

x x-Mittelpunkt des Kreises

y y-Mittelpunkt des Kreises

radius der Radius des Kreises

RÜCKGABEWERTE

status Statusinformation

14.4 page:Clip

BEZEICHNUNG

page:Clip – ändert den Beschneidungspfad

ÜBERSICHT

status = page:Clip()

BESCHREIBUNG

page:Clip() ändert den aktuellen Beschneidungspfad, indem er ihn mit dem aktuellen Pfad unter Verwendung der Nicht-Null-Windungszahl-Regel schneidet. Der Beschneidungspfad wird nur nach dem erfolgreichen Zeichnungsvorgang geändert. Um den aktuellen Pfad nicht weiter zu zeichnen, verwenden Sie den Befehl `page:EndPath()`.

Die folgenden Zeichnungsvorgänge wirken sich nur auf die Bereiche der Seite aus, die im Beschneidungspfad enthalten sind. Zu Beginn enthält der Beschneidungspfad die gesamte Seite. Es gibt keine Möglichkeit, den aktuellen Beschneidungspfad zu vergrößern oder ihn durch einen neuen zu ersetzen. Die Befehle `page:GSave()` und `page:GRestore()` können verwendet werden, um den aktuellen Grafikzustand einschließlich des Beschneidungspfades zu speichern und wiederherzustellen.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.5 page:ClosePath

BEZEICHNUNG

page:ClosePath – schließt den Unterpfad

ÜBERSICHT

```
status = page:ClosePath()
```

BESCHREIBUNG

page:ClosePath() fügt eine gerade Linie vom aktuellen Punkt zum Startpunkt des Unterpfads hinzu. Der aktuelle Punkt ist nun der Startpunkt des Unterpfads.

EINGABEN

keine

RÜCKGABEWERTE

```
status    Statusinformation
```

14.6 page:ClosePathEofillStroke

BEZEICHNUNG

page:ClosePathEofillStroke – schließt, füllt mit Gerade-Ungerade-Regel und zeichnet den Pfad

ÜBERSICHT

```
status = page:ClosePathEofillStroke()
```

BESCHREIBUNG

page:ClosePathEofillStroke() schließt den aktuellen Pfad, füllt den aktuellen Pfad mit der Gerade-Ungerade-Regel und zeichnet anschließend den Pfad.

EINGABEN

keine

RÜCKGABEWERTE

```
status    Statusinformation
```

14.7 page:ClosePathFillStroke

BEZEICHNUNG

page:ClosePathFillStroke – schließt, füllt mit Windungszahl-Algorithmus und zeichnet den Pfad

ÜBERSICHT

```
status = page:ClosePathFillStroke()
```

BESCHREIBUNG

page:ClosePathFillStroke() schließt den aktuellen Pfad, füllt ihn unter Verwendung der Regel mit der Windungszahl ungleich null und zeichnet den Pfad.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.8 page:ClosePathStroke**BEZEICHNUNG**

page:ClosePathStroke – schließt und zeichnet den Pfad

ÜBERSICHT

status = page:ClosePathStroke()

BESCHREIBUNG

page:ClosePathStroke() schließt den aktuellen Pfad und dann wird der Pfad gezeichnet.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.9 page:Concat**BEZEICHNUNG**

page:Concat – verknüpft Matrizen

ÜBERSICHT

status = page:Concat(a, b, c, d, x, y)

BESCHREIBUNG

page:Concat() verknüpft die aktuelle Transformationsmatrix der Seite mit der angegebenen Matrix.

Wenn Sie beispielsweise das Koordinatensystem der Seite um 45 Grad drehen möchten, verwenden Sie page:Concat() wie folgt:

```
Local rad1 = 45 / 180 * #PI
page:Concat(Cos(rad1),Sin(rad1),-Sin(rad1),Cos(rad1),220,350)
```

Um das Koordinatensystem der Seite auf 300 dpi zu ändern, verwenden Sie page:Concat() wie folgt:

```
page:Concat(72.0 / 300.0, 0, 0, 72.0 / 300.0, 0, 0)
```

Rufen Sie page:GSave() vor page:Concat() auf. Dann kann die Änderung durch page:Concat() durch Aufruf von page:GRestore() wiederhergestellt werden.

```
; speichert die aktuellen Grafikzustände
page:GSave(page)
```

```

; verknüpft die Transformationsmatrix
page:Concat(72.0 / 300.0, 0, 0, 72.0 / 300.0, 0, 0)

; zeigt den Text bei den verschobenen Koordinaten an
page:BeginText()
page:MoveTextPos(50, 100)
page:ShowText("Text on the translated coordinates")
page:EndText(page)

; stellt die Grafikzustände wieder her
page:GRestore()

```

Eine Anwendung kann `page:GSave()` aufrufen, wenn sich der Grafikmodus der Seite in `#HPDF_GMODE_PAGE_DESCRIPTION` befindet.

Liefert `#HPDF_OK` bei Erfolg. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

- a Skalierung der x-Koordinate (Größenveränderung)
- b Rotation der x-Koordinate (Drehung)
- c Rotation der y-Koordinate (Drehung)
- d Skalierung der y-Koordinate (Größenveränderung)
- x Translation der x-Koordinate (Verschiebung)
- y Translation der y-Koordinate (Verschiebung)

RÜCKGABEWERTE

`status` Statusinformation

14.10 `page:CreateCircleAnnot`

BEZEICHNUNG

`page:CreateCircleAnnot` – erstellt ein Kreis-Anmerkungsobjekt

ÜBERSICHT

```
ant = page:CreateCircleAnnot(rect, text, encoder)
```

BESCHREIBUNG

`page:CreateCircleAnnot()` erstellt ein neues Kreis-Anmerkungsobjekt für die Seite. Der Parameter `rect` muss eine Tabelle sein, die die Felder `left`, (`links`), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

- `rect` ein Rechteck des anklickbaren Bereichs
- `text` der anzuzeigende Text
- `encoder` eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie `Nil` ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

`ant` Kennung des Anmerkungsobjekt

FEHLER

`#HPDF_INVALID_PAGE` - Es wurde eine ungültige Seiten-Kennung gesetzt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_INVALID_ENCODER` - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.11 page:CreateDestination**BEZEICHNUNG**

`page:CreateDestination` – erstellt ein Zielobjekt

ÜBERSICHT

`dst = page:CreateDestination()`

BESCHREIBUNG

`page:CreateDestination()` erstellt ein neues Zielobjekt für die Seite.

EINGABEN

keine

RÜCKGABEWERTE

`dst` Kennung des Zielobjekts

FEHLER

`#HPDF_INVALID_PAGE` - Eine ungültige Seiten-Kennung wurde gesetzt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

14.12 page:CreateFreeTextAnnot**BEZEICHNUNG**

`page:CreateFreeTextAnnot` – erstellt ein Freier Text-Anmerkungsobjekt

ÜBERSICHT

`ant = page:CreateFreeTextAnnot(rect, text, encoder)`

BESCHREIBUNG

`page:CreateFreeTextAnnot()` erstellt ein neues Freier Text-Anmerkungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

`rect` ein Rechteck des anklickbaren Bereichs

`text` der anzuzeigende Text

`encoder` eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie `Nil` ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

`ant` Kennung des Anmerkungsobjekt

FEHLER

`#HPDF_INVALID_PAGE` - Eine ungültige Seiten-Kennung wurde gesetzt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_INVALID_ENCODER` - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.13 page:CreateHighlightAnnot**BEZEICHNUNG**

`page:CreateHighlightAnnot` – erstellt ein Hervorhebungs-Anmerkungsobjekt

ÜBERSICHT

`ant = page:CreateHighlightAnnot(rect, text, encoder)`

BESCHREIBUNG

`page:CreateHighlightAnnot()` erstellt ein neues Hervorhebungs-Anmerkungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

`rect` ein Rechteck des anklickbaren Bereichs

`text` der anzuzeigende Text

`encoder` eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie `Nil` ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

`ant` Kennung des Anmerkungsobjekt

FEHLER

`#HPDF_INVALID_PAGE` - Eine ungültige Seiten-Kennung wurde gesetzt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_INVALID_ENCODER` - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.14 page:CreateLineAnnot**BEZEICHNUNG**

`page:CreateLineAnnot` – erstellt ein Linien-Anmerkungsobjekt

ÜBERSICHT

`ant = page:CreateLineAnnot(text, encoder)`

BESCHREIBUNG

`page:CreateLineAnnot()` erstellt ein neues Linien-Anmerkungsobjekt für die Seite.

EINGABEN

- text** der anzuzeigende Text
- encoder** eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird;
wenn sie Nil ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

- ant** Kennung des Anmerkungsobjekt

FEHLER

- #HPDF_INVALID_PAGE - Eine ungültige Seiten-Kennung wurde gesetzt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_INVALID_ENCODER - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.15 page:CreateLinkAnnot**BEZEICHNUNG**

page:CreateLinkAnnot – erstellt ein Verknüpfungs-Anmerkungsobjekt

ÜBERSICHT

ant = page:CreateLinkAnnot(**rect**, **dst**)

BESCHREIBUNG

page:CreateLinkAnnot() erstellt ein neues Verknüpfungs-Anmerkungsobjekt für die Seite.

Der Parameter **rect** muss eine Tabelle sein, die die Felder **left** (links), **top** (oben), **right** (rechts) und **bottom** (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

- rect** ein Rechteck des anklickbaren Bereichs
- dst** eine Kennung des Zielobjekts, zu dem gesprungen werden soll

RÜCKGABEWERTE

- ant** Kennung des Anmerkungsobjekt

FEHLER

- #HPDF_INVALID_PAGE - Eine ungültige Seiten-Kennung wurde gesetzt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_INVALID_DESTINATION - Es wurde eine ungültige Ziel-Kennung angegeben.

14.16 page:CreatePopupAnnot**BEZEICHNUNG**

page:CreatePopupAnnot – erstellt ein Notizfenster-Anmerkungsobjekt

ÜBERSICHT

ant = page:CreatePopupAnnot(**rect**, **parent**)

BESCHREIBUNG

`page:CreatePopupAnnot()` erstellt ein neues Notizfenster-Anmerknungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

`rect` ein Rechteck des anklickbaren Bereichs

`parent` übergeordnetes Anmerknungsobjekt

RÜCKGABEWERTE

`ant` Kennung des Anmerknungsobjekt

14.17 page:CreateProjectionAnnot**BEZEICHNUNG**

`page:CreateProjectionAnnot` – erstellt ein Projektions-Anmerknungsobjekt

ÜBERSICHT

`ant = page:CreateProjectionAnnot(rect, text, encoder)`

BESCHREIBUNG

`page:CreateProjectionAnnot()` erstellt ein neues Projektions-Anmerknungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

`rect` ein Rechteck des anklickbaren Bereichs

`text` der anzuzeigende Text

`encoder` eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie `Nil` ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

`ant` Kennung des Anmerknungsobjekt

FEHLER

`#HPDF_INVALID_PAGE` - Eine ungültige Seiten-Kennung wurde gesetzt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_INVALID_ENCODER` - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.18 page:CreateSquareAnnot

BEZEICHNUNG

page:CreateSquareAnnot – erstellt ein Quadrat-Anmerknungsobjekt

ÜBERSICHT

ant = page:CreateSquareAnnot(rect, text, encoder)

BESCHREIBUNG

page:CreateSquareAnnot() erstellt ein neues Quadrat-Anmerknungsobjekt für die Seite.

Der Parameter **rect** muss eine Tabelle sein, die die Felder **left** (links), **top** (oben), **right** (rechts) und **bottom** (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

rect ein Rechteck des anklickbaren Bereichs

text der anzuzeigende Text

encoder eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie Nil ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

ant Kennung des Anmerknungsobjekt

FEHLER

#HPDF_INVALID_PAGE - Eine ungültige Seiten-Kennung wurde gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

#HPDF_INVALID_ENCODER - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.19 page:CreateSquigglyAnnot

BEZEICHNUNG

page:CreateSquigglyAnnot – erstellt ein Unterringelungs-Anmerknungsobjekt

ÜBERSICHT

ant = page:CreateSquigglyAnnot(rect, text, encoder)

BESCHREIBUNG

page:CreateSquigglyAnnot() erstellt ein neues Unterringelungs-Anmerknungsobjekt für die Seite.

Der Parameter **rect** muss eine Tabelle sein, die die Felder **left** (links), **top** (oben), **right** (rechts) und **bottom** (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

rect ein Rechteck des anklickbaren Bereichs

text der anzuzeigende Text

encoder eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie Nil ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

`ant` Kennung des Anmerkungsobjekt

FEHLER

`#HPDF_INVALID_PAGE` - Eine ungültige Seiten-Kennung wurde gesetzt.

`#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.

`#HPDF_INVALID_ENCODER` - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.20 page:CreateStampAnnot**BEZEICHNUNG**

`page:CreateStampAnnot` – erstellt ein Stempel-Anmerkungsobjekt

ÜBERSICHT

`ant = page:CreateStampAnnot(rect, stamp, text, encoder)`

BESCHREIBUNG

`page:CreateStampAnnot()` erstellt ein neues Stempel-Anmerkungsobjekt für die Seite.

Der Parameter `stamp` muss eine der folgenden Konstanten sein:

```
#HPDF_STAMP_ANNOT_APPROVED
#HPDF_STAMP_ANNOT_EXPERIMENTAL
#HPDF_STAMP_ANNOT_NOTAPPROVED
#HPDF_STAMP_ANNOT_ASIS
#HPDF_STAMP_ANNOT_EXPIRED
#HPDF_STAMP_ANNOT_NOTFORPUBLICRELEASE
#HPDF_STAMP_ANNOT_CONFIDENTIAL
#HPDF_STAMP_ANNOT_FINAL
#HPDF_STAMP_ANNOT_SOLD
#HPDF_STAMP_ANNOT_DEPARTMENTAL
#HPDF_STAMP_ANNOT_FORCOMMENT
#HPDF_STAMP_ANNOT_TOPSECRET
#HPDF_STAMP_ANNOT_DRAFT
#HPDF_STAMP_ANNOT_FORPUBLICRELEASE
```

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

`rect` ein Rechteck des anklickbaren Bereichs

`stamp` Stempel-Anmerkungstyp (mögliche Werte siehe oben)

`text` der anzuzeigende Text

`encoder` eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie `Nil` ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

`ant` Kennung des Anmerkungsobjekt

FEHLER

- #HPDF_INVALID_PAGE - Eine ungültige Seiten-Kennung wurde gesetzt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_INVALID_ENCODER - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.21 page:CreateStrikeOutAnnot**BEZEICHNUNG**

page:CreateStrikeOutAnnot – erstellt ein Durchstreich-Anmerkungsbjekt

ÜBERSICHT

```
ant = page:CreateStrikeOutAnnot(rect, text, encoder)
```

BESCHREIBUNG

page:CreateStrikeOutAnnot() erstellt ein neues Durchstreich-Anmerkungsbjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

- `rect` ein Rechteck des anklickbaren Bereichs
- `text` der anzuzeigende Text
- `encoder` eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie `Nil` ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

- `ant` Kennung des Anmerkungsbjekt

FEHLER

- #HPDF_INVALID_PAGE - Eine ungültige Seiten-Kennung wurde gesetzt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_INVALID_ENCODER - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.22 page:CreateTextAnnot**BEZEICHNUNG**

page:CreateTextAnnot – erstellt ein Text-Anmerkungsbjekt

ÜBERSICHT

```
ant = page:CreateTextAnnot(rect, text, encoder)
```

BESCHREIBUNG

page:CreateTextAnnot() erstellt ein neues Text-Anmerkungsbjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

<code>rect</code>	ein Rechteck des anklickbaren Bereichs
<code>text</code>	der anzuzeigende Text
<code>encoder</code>	eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie <code>Nil</code> ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

<code>ant</code>	Kennung des Anmerkungsobjekt
------------------	------------------------------

FEHLER

- #HPDF_INVALID_PAGE - Eine ungültige Seiten-Kennung wurde gesetzt.
- #HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.
- #HPDF_INVALID_ENCODER - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.23 page:CreateTextMarkupAnnot

BEZEICHNUNG

`page:CreateTextMarkupAnnot` – erstellt ein Textmarkup-Anmerkungsobjekt

ÜBERSICHT

`ant = page:CreateTextMarkupAnnot(rect, text, encoder, subtype)`

BESCHREIBUNG

`page:CreateTextMarkupAnnot()` erstellt ein neues Textmarkup-Anmerkungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

Der Parameter `subtype` muss eine der folgenden Konstanten sein:

```
#HPDF_ANNOT_TEXT_NOTES
#HPDF_ANNOT_LINK
#HPDF_ANNOT_SOUND
#HPDF_ANNOT_FREE_TEXT
#HPDF_ANNOT_STAMP
#HPDF_ANNOT_SQUARE
#HPDF_ANNOT_CIRCLE
#HPDF_ANNOT_STRIKE_OUT
#HPDF_ANNOT_HIGHLIGHT
#HPDF_ANNOT_UNDERLINE
#HPDF_ANNOT_INK
#HPDF_ANNOT_FILE_ATTACHMENT
#HPDF_ANNOT_POPUP
#HPDF_ANNOT_3D
#HPDF_ANNOT_SQUIGGLY
#HPDF_ANNOT_LINE
#HPDF_ANNOT_PROJECTION
#HPDF_ANNOT_WIDGET
```

EINGABEN

<code>rect</code>	ein Rechteck des anklickbaren Bereichs
<code>text</code>	der anzuzeigende Text
<code>encoder</code>	eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie <code>Nil</code> ist, wird die Standardkodierung verwendet.
<code>subtype</code>	Untertyp des Anmerkungsobjekt (mögliche Werte siehe oben)

RÜCKGABEWERTE

<code>ant</code>	Kennung des Anmerkungsobjekt
------------------	------------------------------

FEHLER

- `#HPDF_INVALID_PAGE` - Eine ungültige Seiten-Kennung wurde gesetzt.
- `#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.
- `#HPDF_INVALID_ENCODER` - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.24 `page:CreateUnderlineAnnot`

BEZEICHNUNG

`page:CreateUnderlineAnnot` – erstellt ein Unterstreichungs-Anmerkungsobjekt

ÜBERSICHT

`ant = page:CreateUnderlineAnnot(rect, text, encoder)`

BESCHREIBUNG

`page:CreateUnderlineAnnot()` erstellt ein neues Unterstreichungs-Anmerkungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

<code>rect</code>	ein Rechteck des anklickbaren Bereichs
<code>text</code>	der anzuzeigende Text
<code>encoder</code>	eine Kodierungs-Kennung, die zum kodieren des Textes verwendet wird; wenn sie <code>Nil</code> ist, wird die Standardkodierung verwendet.

RÜCKGABEWERTE

<code>ant</code>	Kennung des Anmerkungsobjekt
------------------	------------------------------

FEHLER

- `#HPDF_INVALID_PAGE` - Eine ungültige Seiten-Kennung wurde gesetzt.
- `#HPDF_FAILED_TO_ALLOC_MEM` - Speicherzuordnung ist fehlgeschlagen.
- `#HPDF_INVALID_ENCODER` - Es wurde eine ungültige Kodierungs-Kennung angegeben.

14.25 page:CreateURILinkAnnot

BEZEICHNUNG

page:CreateURILinkAnnot – erstellt ein Web-Link-Anmerknungsobjekt

ÜBERSICHT

```
ant = page:CreateURILinkAnnot(rect, uri)
```

BESCHREIBUNG

page:CreateURILinkAnnot() erstellt ein neues Web-Link-Anmerknungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

`rect` ein Rechteck des anklickbaren Bereichs

`uri` URL des Ziels, zu dem gesprungen werden soll

RÜCKGABEWERTE

`ant` Kennung des Anmerknungsobjekt

FEHLER

#HPDF_INVALID_PAGE - Eine ungültige Seiten-Kennung wurde gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

14.26 page:CreateWidgetAnnot

BEZEICHNUNG

page:CreateWidgetAnnot – erstellt ein Formular-Anmerknungsobjekt

ÜBERSICHT

```
ant = page:CreateWidgetAnnot(rect)
```

BESCHREIBUNG

page:CreateWidgetAnnot() erstellt ein neues Formular-Anmerknungsobjekt für die Seite.

Der Parameter `rect` muss eine Tabelle sein, die die Felder `left` (links), `top` (oben), `right` (rechts) und `bottom` (unten) enthält, die ein Rechteck beschreiben.

EINGABEN

`rect` ein Rechteck des anklickbaren Bereichs

RÜCKGABEWERTE

`ant` Kennung des Anmerknungsobjekt

14.27 page:CurveTo

BEZEICHNUNG

page:CurveTo – hängt eine Bezierkurve an den Pfad an

ÜBERSICHT

status = page:CurveTo(x1, y1, x2, y2, x3, y3)

BESCHREIBUNG

page:CurveTo() fügt dem aktuellen Pfad eine Bezierkurve mit den Kontrollpunkten (x1, y1) und (x2, y2) und (x3, y3) hinzu und setzt dann den aktuellen Punkt auf (x3, y3).

EINGABEN

x1	x-Koordinate des Kontrollpunkts #1
y1	y-Koordinate des Kontrollpunkts #1
x2	x-Koordinate des Kontrollpunkts #2
y2	y-Koordinate des Kontrollpunkts #2
x3	x-Koordinate des Kurvenzielpunktes
y3	y-Koordinate des Kurvenzielpunktes

RÜCKGABEWERTE

status	Statusinformation
--------	-------------------

14.28 page:CurveTo2

BEZEICHNUNG

page:CurveTo2 – hängt eine Bezierkurve an den Pfad an

ÜBERSICHT

status = page:CurveTo2(x2, y2, x3, y3)

BESCHREIBUNG

page:CurveTo2() hängt eine Bezierkurve an den aktuellen Pfad an, wobei der aktuelle Punkt und (x2, y2) und (x3, y3) als Kontrollpunkte verwendet werden. Dann wird der aktuelle Punkt auf (x3, y3) gesetzt.

EINGABEN

x2	x-Koordinate des Kontrollpunkts #1
y2	y-Koordinate des Kontrollpunkts #1
x3	x-Koordinate des Kontrollpunkts #2
y3	y-Koordinate des Kontrollpunkts #2

RÜCKGABEWERTE

status	Statusinformation
--------	-------------------

14.29 page:CurveTo3

BEZEICHNUNG

page:CurveTo3 – hängt eine Bezierkurve an den Pfad an

ÜBERSICHT

```
status = page:CurveTo3(x1, y1, x3, y3)
```

BESCHREIBUNG

page:CurveTo3() fügt dem aktuellen Pfad eine Bezierkurve mit den zwei angegebenen Punkten hinzu. Der Punkt (x1, y1) und der Punkt (x3, y3) werden als Kontrollpunkte für eine Bezierkurve verwendet und der aktuelle Punkt wird auf den Punkt (x3, y3) verschoben.

EINGABEN

x1	x-Koordinate des Kontrollpunkts #1
y1	y-Koordinate des Kontrollpunkts #1
x3	x-Koordinate des Kontrollpunkts #2
y3	y-Koordinate des Kontrollpunkts #2

RÜCKGABEWERTE

status	Statusinformation
--------	-------------------

14.30 page:DrawImage

BEZEICHNUNG

page:DrawImage – zeichnet ein Bild auf die Seite

ÜBERSICHT

```
status = page:DrawImage(image, x, y, width, height)
```

BESCHREIBUNG

page:DrawImage() zeigt ein Bild in einem Arbeitsgang an.

EINGABEN

image	die Kennung eines Bildobjekts
x	horizontale Koordinate für das Bild
y	vertikale Koordinate für das Bild
width	die Breite des Bereichs, in dem das Bild angezeigt werden soll
height	die Höhe des Bereichs, in dem das Bild angezeigt werden soll

RÜCKGABEWERTE

status	Statusinformation
--------	-------------------

14.31 page:Ellipse

BEZEICHNUNG

page:Ellipse – hängt eine Ellipse an den Pfad an

ÜBERSICHT

```
status = page:Ellipse(x, y, xradius, yradius)
```

BESCHREIBUNG

page:Ellipse() hängt eine Ellipse an den aktuellen Pfad an.

EINGABEN

x	x-Mittelpunkt der Ellipse
y	y-Mittelpunkt der Ellipse
xradius	horizontaler Radius der Ellipse
yradius	vertikaler Radius der Ellipse

RÜCKGABEWERTE

status Statusinformation

14.32 page:EndPath

BEZEICHNUNG

page:EndPath – beendet den Pfad

ÜBERSICHT

```
status = page:EndPath()
```

BESCHREIBUNG

page:EndPath() beendet das Pfadobjekt, ohne zu füllen oder zu zeichnen.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.33 page:EndText

BEZEICHNUNG

page:EndText – beendet ein Textobjekt

ÜBERSICHT

```
status = page:EndText()
```

BESCHREIBUNG

page:EndText() beendet ein Textobjekt.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

14.34 page:EoClip**BEZEICHNUNG**

`page:EoClip` – ändert den Beschneidungspfad mit der Gerade-Ungerade-Regel

ÜBERSICHT

`status = page:EoClip()`

BESCHREIBUNG

`page:Clip()` ändert den aktuellen Beschneidungspfad, indem er ihn mit dem aktuellen Pfad unter Verwendung der Gerade-Ungerade-Regel schneidet. Der Beschneidungspfad wird nur nach dem erfolgreichen Zeichnungsvorgang geändert. Um den aktuellen Pfad nicht weiter zu zeichnen, verwenden Sie den Befehl `page:EndPath()`.

Die folgenden Zeichnungsvorgänge wirken sich nur auf die Bereiche der Seite aus, die im Beschneidungspfad enthalten sind. Zu Beginn enthält der Beschneidungspfad die gesamte Seite. Es gibt keine Möglichkeit, den aktuellen Beschneidungspfad zu vergrößern oder ihn durch einen neuen zu ersetzen. Die Befehle `page:GSave()` und `page:GRestore()` können verwendet werden, um den aktuellen Grafikzustand einschließlich des Beschneidungspfades zu speichern und wiederherzustellen.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

14.35 page:Eofill**BEZEICHNUNG**

`page:Eofill` – füllt den aktuellen Pfad mit der Gerade-Ungerade-Regel

ÜBERSICHT

`status = page:Eofill()`

BESCHREIBUNG

`page:Eofill()` füllt den aktuellen Pfad mit der Gerade-Ungerade-Regel.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

14.36 page:EofillStroke

BEZEICHNUNG

page:EofillStroke – füllt und zeichnet den aktuellen Pfad mit der Gerade-Ungerade-Regel

ÜBERSICHT

```
status = page:EofillStroke()
```

BESCHREIBUNG

page:EofillStroke() füllt den aktuellen Pfad mit der Gerade-Ungerade-Regel, dann wird der Pfad gezeichnet.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.37 page:ExecuteXObject

BEZEICHNUNG

page:ExecuteXObject – führt das XObjekt aus

ÜBERSICHT

```
status = page:ExecuteXObject(xobj)
```

BESCHREIBUNG

page:ExecuteXObject() führt das angegebene XObjekt aus.

EINGABEN

xobj Kennung des XObjekt

RÜCKGABEWERTE

status Statusinformation

14.38 page:Fill

BEZEICHNUNG

page:Fill – füllt den aktuellen Pfad mit dem Windungszahl-Algorithmus

ÜBERSICHT

```
status = page:Fill()
```

BESCHREIBUNG

page:Fill() füllt den aktuellen Pfad unter Verwendung der Regel mit der Windungszahl ungleich null.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.39 page:FillStroke

BEZEICHNUNG

page:FillStroke – füllt und zeichnet den aktuellen Pfad mit dem Windungszahl-Algorithmus

ÜBERSICHT

```
status = page:FillStroke()
```

BESCHREIBUNG

page:FillStroke() füllt den aktuellen Pfad unter Verwendung der Regel mit der Windungszahl ungleich Null, dann wird der Pfad gezeichnet.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.40 page:GetCharSpace

BEZEICHNUNG

page:GetCharSpace – ermittelt den aktuellen Zeichenabstand

ÜBERSICHT

```
charspace = page:GetCharSpace()
```

BESCHREIBUNG

page:GetCharSpace() ermittelt den aktuellen Wert des Zeichenabstandes der Seite.

EINGABEN

keine

RÜCKGABEWERTE

charspace
der aktuelle Zeichenabstand

14.41 page:GetCMYKFill

BEZEICHNUNG

page:GetCMYKFill – ermittelt die CMYK-Füllfarbe

ÜBERSICHT

```
t = page:GetCMYKFill()
```

BESCHREIBUNG

page:GetCMYKFill() gibt den aktuellen Wert der Füllfarbe der Seite zurück. page:GetCMYKFill() ist nur gültig, wenn der Füllfarbenraum der Seite #HPDF_CS_DEVICE_CMYK ist.

Dieser Befehl gibt eine Tabelle mit den folgenden initialisierten Feldern zurück:

C Grünblau-Anteil der Farbe (Cyan)

M	Magentarot-Anteil der Farbe (violett/magenta)
Y	Gelb-Anteil der Farbe (Yellow)
K	Schwarz-Anteil der Farbe (Key/Black)

Alle Felder enthalten Werte zwischen 0 und 1.

EINGABEN

keine

RÜCKGABEWERTE

t die aktuelle CMYK-Füllfarbe

14.42 page:GetCMYKStroke

BEZEICHNUNG

page:GetCMYKStroke – ermittelt die aktuelle CMYK-Zeichnungsfarbe

ÜBERSICHT

t = page:GetCMYKStroke()

BESCHREIBUNG

page:GetCMYKStroke() gibt den aktuellen Wert der Zeichnungsfarbe der Seite zurück.
page:GetCMYKStroke() ist nur gültig, wenn der Farbraum der Seite #HPDF_CS_DEVICE_CMYK ist.

Dieser Befehl gibt eine Tabelle mit den folgenden initialisierten Feldern zurück:

C	Grünblau-Anteil der Farbe (Cyan)
M	Magentarot-Anteil der Farbe (violett/magenta)
Y	Gelb-Anteil der Farbe (Yellow)
K	Schwarz-Anteil der Farbe (Key/Black)

Alle Felder enthalten Werte zwischen 0 und 1.

EINGABEN

keine

RÜCKGABEWERTE

t die aktuelle CMYK-Zeichnungsfarbe

14.43 page:GetCurrentFont

BEZEICHNUNG

page:GetCurrentFont – ermittelt die aktuelle Schriftart

ÜBERSICHT

font = page:GetCurrentFont()

BESCHREIBUNG

page:GetCurrentFont() ermittelt die Kennung der aktuellen Schriftart der Seite.

EINGABEN

keine

RÜCKGABEWERTE

font Kennung der Schriftart

14.44 page:GetCurrentFontSize

BEZEICHNUNG

page:GetCurrentFontSize – ermittelt die aktuelle Schriftgröße

ÜBERSICHT

```
size = page:GetCurrentFontSize()
```

BESCHREIBUNG

page:GetCurrentFontSize() ermittelt die Größe der aktuellen Schriftart der Seite.

EINGABEN

keine

RÜCKGABEWERTE

size die aktuelle Schriftgröße

14.45 page:GetCurrentPos

BEZEICHNUNG

page:GetCurrentPos – ermittelt die aktuelle Pfadposition

ÜBERSICHT

```
x, y = page:GetCurrentPos()
```

BESCHREIBUNG

page:GetCurrentPos() ermittelt die aktuelle Position für das Pfadzeichnen.

Ein Programm kann page:GetCurrentPos() nur aufrufen, wenn der Grafikmodus #HPDF_GMODE_PATH_OBJECT ist.

EINGABEN

keine

RÜCKGABEWERTE

x aktuelle x-Position

y aktuelle y-Position

14.46 page:GetCurrentTextPos

BEZEICHNUNG

page:GetCurrentTextPos – ermittelt die aktuelle Textposition

ÜBERSICHT

x, y = page:GetCurrentTextPos()

BESCHREIBUNG

page:GetCurrentTextPos() ermittelt die aktuelle Position zum Zeichnen von Text.

Ein Programm kann page:GetCurrentTextPos() nur aufrufen, wenn der Grafikmodus #HPDF_GMODE_TEXT_OBJECT ist.

EINGABEN

keine

RÜCKGABEWERTE

x aktuelle x-Position

y aktuelle y-Position

14.47 page:GetDash

BEZEICHNUNG

page:GetDash – ermittelt das aktuelle Strichmuster

ÜBERSICHT

t = page:GetDash()

BESCHREIBUNG

page:GetDash() ermittelt das aktuelle Strichmuster der Seite.

Diese Methode gibt eine Tabelle mit den folgenden initialisierten Feldern zurück:

ptn Eine Tabelle, die die einzelnen Linienstücke und Lücken des Musters enthält.

num_ptn Die Anzahl der Elemente in der Tabelle ptn.

phase Die Phase, in der das Muster beginnt.

Siehe [Abschnitt 14.81 \[page:SetDash\]](#), Seite 151, für Details.

EINGABEN

keine

RÜCKGABEWERTE

t Tabelle mit dem aktuellen Strichmuster (siehe oben)

14.48 page:GetFillingColorSpace

BEZEICHNUNG

page:GetFillingColorSpace – ermittelt den ausfüllenden Farbraum

ÜBERSICHT

```
cs = page:GetFillingColorSpace()
```

BESCHREIBUNG

page:GetFillingColorSpace() gibt den aktuellen Wert des Füllfarbenraums der Seite zurück. Dies ist eine Konstante von #HPDF_CS_DEVICE_GRAY, #HPDF_CS_DEVICE_RGB oder #HPDF_CS_DEVICE_CMYK.

EINGABEN

keine

RÜCKGABEWERTE

cs aktueller Füllfarbraum

14.49 page:GetFlat

BEZEICHNUNG

page:GetFlat – ermittelt den aktuellen Kurvennäherungs-Wert

ÜBERSICHT

```
flat = page:GetFlat()
```

BESCHREIBUNG

page:GetFlat() ermittelt den aktuellen Kurvennäherungs-Wert der Seite.

EINGABEN

keine

RÜCKGABEWERTE

flat aktuellen Kurvennäherungs-Wert

14.50 page:GetGMode

BEZEICHNUNG

page:GetGMode – ermittelt den aktuellen Grafikmodus

ÜBERSICHT

```
mode = page:GetGMode()
```

BESCHREIBUNG

page:GetGMode() ermittelt den aktuellen Grafikmodus.

Die folgenden Grafikmodi stehen zur Verfügung:

```
#HPDF_GMODE_PAGE_DESCRIPTION  
#HPDF_GMODE_PATH_OBJECT  
#HPDF_GMODE_TEXT_OBJECT
```

```
#HPDF_GMODE_CLIPPING_PATH
#HPDF_GMODE_SHADING
#HPDF_GMODE_INLINE_IMAGE
#HPDF_GMODE_EXTERNAL_OBJECT
```

EINGABEN

keine

RÜCKGABEWERTE

mode aktueller Grafikmodus

14.51 page:GetGrayFill**BEZEICHNUNG**

page:GetGrayFill – ermittelt die graue Füllfarbe

ÜBERSICHT

gray = page:GetGrayFill()

BESCHREIBUNG

page:GetGrayFill() gibt den aktuellen Wert der Füllfarbe der Seite zurück. page:GetGrayFill() ist nur gültig, wenn der Füllfarbenraum der Seite #HPDF_CS_DEVICE_GRAY ist.

EINGABEN

keine

RÜCKGABEWERTE

gray aktuelle graue Füllfarbe

14.52 page:GetGrayStroke**BEZEICHNUNG**

page:GetGrayStroke – ermittelt die graue Zeichnungsfarbe

ÜBERSICHT

gray = page:GetGrayStroke()

BESCHREIBUNG

page:GetGrayStroke() gibt den aktuellen Wert der Strichfarbe der Seite zurück. page:GetGrayStroke() ist nur gültig, wenn der Farbraum der Seite #HPDF_CS_DEVICE_GRAY ist.

EINGABEN

keine

RÜCKGABEWERTE

gray die graue Zeichnungsfarbe

14.53 page:GetGStateDepth

BEZEICHNUNG

page:GetGStateDepth – ermittelt den Grafikstatusstapel

ÜBERSICHT

d = page:GetGStateDepth()

BESCHREIBUNG

page:GetGStateDepth() gibt die Nummer des Grafikstatusstapels der Seite zurück.

EINGABEN

keine

RÜCKGABEWERTE

d aktueller Grafikstatusstapel

14.54 page:GetHeight

BEZEICHNUNG

page:GetHeight – ermittelt die Seitenhöhe

ÜBERSICHT

h = page:GetHeight()

BESCHREIBUNG

page:GetHeight() ermittelt die Höhe der Seite.

EINGABEN

keine

RÜCKGABEWERTE

h Seitenhöhe

14.55 page:GetHorizontalScaling

BEZEICHNUNG

page:GetHorizontalScaling – ermittelt die aktuelle horizontale Skalierung

ÜBERSICHT

s = page:GetHorizontalScaling()

BESCHREIBUNG

page:GetHorizontalScaling() gibt den aktuellen Wert der horizontalen Skalierung der Seite für den Text zurück.

EINGABEN

keine

RÜCKGABEWERTE

s horizontaler Skalierungswert

14.56 `page:GetLineCap`

BEZEICHNUNG

`page:GetLineCap` – ermittelt den aktuellen Stil der Linienenden

ÜBERSICHT

`cap = page:GetLineCap()`

BESCHREIBUNG

`page:GetLineCap()` ermittelt den aktuellen Stil der Linienenden der Seite.

Siehe [Abschnitt 14.89](#) [`page:SetLineCap`], [Seite 154](#), für eine Liste der verfügbaren Linienendenstile.

EINGABEN

keine

RÜCKGABEWERTE

`cap` aktueller Linienendstil

14.57 `page:GetLineJoin`

BEZEICHNUNG

`page:GetLineJoin` – ermittelt den Stil beim Verbinden von Linien

ÜBERSICHT

`linejoin = page:GetLineJoin()`

BESCHREIBUNG

`page:GetLineJoin()` ermittelt den aktuellen Stil beim Verbinden von Linien.

Siehe [Abschnitt 14.90](#) [`page:SetLineJoin`], [Seite 154](#), für eine Liste der verfügbaren Linienverbindungsstile.

EINGABEN

keine

RÜCKGABEWERTE

`linejoin` aktueller Linienverbindungsstil

14.58 `page:GetLineWidth`

BEZEICHNUNG

`page:GetLineWidth` – ermittelt die Linienbreite der Seite

ÜBERSICHT

`w = page:GetLineWidth()`

BESCHREIBUNG

`page:GetLineWidth()` ermittelt die aktuelle Linienbreite der Seite.

EINGABEN

keine

RÜCKGABEWERTE

w aktuelle Linienbreite

14.59 page:GetMiterLimit**BEZEICHNUNG**

page:GetMiterLimit – ermittelt die aktuelle Gehrungsgrenze

ÜBERSICHT

```
limit = page:GetMiterLimit()
```

BESCHREIBUNG

page:GetMiterLimit() ermittelt den aktuellen Wert der Gehrungsgrenze der Seite.

EINGABEN

keine

RÜCKGABEWERTE

limit aktuelle Gehrungsgrenze

14.60 page:GetRGBFill**BEZEICHNUNG**

page:GetRGBFill – ermittelt die aktuelle RGB-Füllfarbe

ÜBERSICHT

```
t = page:GetRGBFill()
```

BESCHREIBUNG

page:GetRGBFill() gibt den aktuellen Wert der Füllfarbe der Seite zurück. page:GetRGBFill() ist nur gültig, wenn der ausfüllende Farbraum der Seite #HPDF_CS_DEVICE_RGB ist.

Dieser Befehl gibt eine Tabelle mit den folgenden initialisierten Feldern zurück:

R	Rot-Wert der Farbe
G	Grün-Wert der Farbe
B	Blau-Wert der Farbe

Alle Felder enthalten Werte zwischen 0 und 1.

EINGABEN

keine

RÜCKGABEWERTE

t aktuelle RGB-Füllfarbe

14.61 page:GetRGBStroke

BEZEICHNUNG

page:GetRGBStroke – ermittelt die RGB-Zeichnungsfarbe

ÜBERSICHT

```
t = page:GetRGBStroke()
```

BESCHREIBUNG

page:GetRGBStroke() gibt den aktuellen Wert der Zeichnungsfarbe der Seite zurück. page:GetRGBStroke() ist nur gültig, wenn der Farbraum der Seite #HPDF_CS_DEVICE_RGB ist.

Dieser Befehl gibt eine Tabelle mit den folgenden initialisierten Feldern zurück:

R	Rot-Wert der Farbe
G	Grün-Wert der Farbe
B	Blau-Wert der Farbe

Alle Felder enthalten Werte zwischen 0 und 1.

EINGABEN

keine

RÜCKGABEWERTE

t aktuelle RGB-Zeichnungsfarbe

14.62 page:GetStrokingColorSpace

BEZEICHNUNG

page:GetStrokingColorSpace – ermittelt den Zeichnungsfarbraum

ÜBERSICHT

```
cs = page:GetStrokingColorSpace()
```

BESCHREIBUNG

page:GetStrokingColorSpace() gibt den aktuellen Wert des Zeichnungsfarbraums der Seite zurück. Dies ist entweder #HPDF_CS_DEVICE_GRAY, #HPDF_CS_DEVICE_RGB oder #HPDF_CS_DEVICE_CMYK.

EINGABEN

keine

RÜCKGABEWERTE

cs aktueller Zeichnungsfarbraum

14.63 page:GetTextLeading

BEZEICHNUNG

page:GetTextLeading – ermittelt den aktuellen Zeilenabstand

ÜBERSICHT

l = page:GetTextLeading()

BESCHREIBUNG

page:GetTextLeading() gibt den aktuellen Wert des Zeilenabstands der Seite zurück.

EINGABEN

keine

RÜCKGABEWERTE

l aktueller Zeilenabstand

14.64 page:GetTextMatrix

BEZEICHNUNG

page:GetTextMatrix – ermittelt die aktuelle Texttransformationsmatrix

ÜBERSICHT

m = page:GetTextMatrix()

BESCHREIBUNG

page:GetTextMatrix() ermittelt die aktuelle Texttransformationsmatrix der Seite.

Diese Methode gibt die Transformationsmatrix in einer Tabelle mit den folgenden initialisierten Feldern zurück:

a	Skalierung der x-Koordinate (Grössenveränderung)
b	Rotation der x-Koordinate (Drehung)
c	Rotation der y-Koordinate (Drehung)
d	Skalierung der y-Koordinate (Grössenveränderung)
x	x - Translation der x-Koordinate (Verschiebung)
y	y - Translation der y-Koordinate (Verschiebung)

EINGABEN

keine

RÜCKGABEWERTE

m aktuelle Texttransformationsmatrix

14.65 page:GetTextRenderingMode

BEZEICHNUNG

page:GetTextRenderingMode – ermittelt den aktuellen Textwiedergabemodus

ÜBERSICHT

mode = page:GetTextRenderingMode()

BESCHREIBUNG

page:GetTextRenderingMode() gibt den aktuellen Wert des Textwiedergabemodus der Seite zurück.

Siehe [Abschnitt 14.100 \[page:SetTextRenderingMode\]](#), Seite 160, für eine Liste der verfügbaren Textwiedergabemodi..

EINGABEN

keine

RÜCKGABEWERTE

mode aktueller Textwiedergabemodus

14.66 page:GetTextRise

BEZEICHNUNG

page:GetTextRise – ermittelt die aktuelle Textlage

ÜBERSICHT

rise = page:GetTextRise()

BESCHREIBUNG

page:GetTextRise() gibt den aktuellen Wert der Textlage der Seite zurück.

EINGABEN

keine

RÜCKGABEWERTE

rise aktuelle Textlage

14.67 page:GetTransMatrix

BEZEICHNUNG

page:GetTransMatrix – ermittelt die aktuelle Transformationsmatrix

ÜBERSICHT

m = page:GetTransMatrix()

BESCHREIBUNG

page:GetTransMatrix() ermittelt die aktuelle Transformationsmatrix der Seite.

Diese Methode gibt die Transformationsmatrix in einer Tabelle mit den folgenden initialisierten Feldern zurück:

a Skalierung der x-Koordinate (Größenveränderung)

b	Rotation der x-Koordinate (Drehung)
c	Rotation der y-Koordinate (Drehung)
d	Skalierung der y-Koordinate (Größenveränderung)
x	x - Translation der x-Koordinate (Verschiebung)
y	y - Translation der y-Koordinate (Verschiebung)

EINGABEN

keine

RÜCKGABEWERTE

m Transformationsmatrix

14.68 page:GetWidth

BEZEICHNUNG

page:GetWidth – ermittelt die Seitenbreite

ÜBERSICHT

w = page:GetWidth()

BESCHREIBUNG

page:GetWidth() ermittelt die Breite der Seite.

EINGABEN

keine

RÜCKGABEWERTE

w Seitenbreite

14.69 page:GetWordSpace

BEZEICHNUNG

page:GetWordSpace – ermittelt den aktuellen Wortabstand

ÜBERSICHT

wordspace = page:GetWordSpace()

BESCHREIBUNG

page:GetWordSpace() gibt den aktuellen Wert des Wortabstandes der Seite zurück.

EINGABEN

keine

RÜCKGABEWERTE

wordspace
aktueller Wortabstand

14.70 page:GRestore

BEZEICHNUNG

page:GRestore – stellt den Grafikzustand wieder her

ÜBERSICHT

status = page:GRestore()

BESCHREIBUNG

page:GRestore() stellt den Grafikzustand wieder her, der durch page:GSave() gespeichert wurde.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.71 page:GSave

BEZEICHNUNG

page:GSave – speichert die aktuellen Grafikparameter

ÜBERSICHT

status = page:GSave()

BESCHREIBUNG

page:GSave() speichert die aktuellen Grafikparameter der Seite. Eine Anwendung kann page:GSave() bis zu 28 Mal aufrufen und die gespeicherten Parameter durch Aufruf von page:GRestore() wiederherstellen.

Die von page:GSave() gespeicherten Parameter sind:

- Beschneidungspfad
- Füllfarbe
- Gehrungslimit
- Horizontale Skalierung
- Kurvennäherung
- Linienbreite
- Linienendenstil
- Linienverbindungsstil
- Schriftart
- Schriftgröße
- Strichmuster
- Textlage
- Transformationsmatrix
- Wiedergabemodus
- Wortabstand

- Zeichenabstand
- Zeichnungsfarbe
- Zeilenabstand

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

14.72 `page:LineTo`

BEZEICHNUNG

`page:LineTo` – hängt eine Linie an den Pfad an

ÜBERSICHT

```
status = page:LineTo(x, y)
```

BESCHREIBUNG

`page:LineTo()` fügt eine Linie vom aktuellen Punkt zum angegebenen Punkt hinzu.

EINGABEN

`x` x-Koordinate des Endpunktes des Pfads
`y` y-Koordinate des Endpunktes des Pfads

RÜCKGABEWERTE

`status` Statusinformation

14.73 `page:MeasureText`

BEZEICHNUNG

`page:MeasureText` – ermittelt die Bytelänge des Textes

ÜBERSICHT

```
bl, rw = page:MeasureText(text, width, wordwrap)
```

BESCHREIBUNG

`page:MeasureText()` berechnet die Bytelänge, die innerhalb der angegebenen Breite enthalten ist.

Der Parameter `wordwrap` legt fest, wie Wörter umgebrochen werden sollen: Angenommen es gibt drei Wörter: "ABCDE", "FGH" und "IJKL". Angenommen die Teilzeichenkette wird so lange verwendet, bis "J" innerhalb der Breite (12 Byte) enthalten sein kann. Wenn `wordwrap` `False` ist, gibt dieser Befehl 12 zurück. Wenn der Parameter `wordwrap` `True` ist, gibt er 10 zurück (das Ende des vorherigen Wortes).

EINGABEN

`text` der Text, dessen Länge berechnet werden soll

`width` die Breite des Bereichs, in dem der Text platziert werden soll
`wordwrap` boolesch, der angibt, ob Zeilenumbrüche verwendet werden sollen.

RÜCKGABEWERTE

`bl` Bytelänge des Textes
`rw` tatsächliche Breite des Textes

14.74 page:MoveTextPos**BEZEICHNUNG**

`page:MoveTextPos` – ändert die aktuelle Textposition

ÜBERSICHT

`status = page:MoveTextPos(x, y[, lead])`

BESCHREIBUNG

`page:MoveTextPos()` ändert unter Verwendung der angegebenen Offsetwerte die aktuelle Textposition. Wenn die aktuelle Textposition (x1, y1) ist, wird die neue Textposition (x1 + x, y1 + y) sein.

Wenn das optionale Argument `lead` auf `True` gesetzt ist, wird der führende Text auf `-y` gesetzt.

EINGABEN

`x` x-Offset für den Text
`y` y-Offset für den Text
`lead` optional: ob Text auf `-y` gesetzt werden soll (`True`) oder nicht (`False`)

RÜCKGABEWERTE

`status` Statusinformation

14.75 page:MoveTo**BEZEICHNUNG**

`page:MoveTo` – startet einen neuen Unterpfad

ÜBERSICHT

`status = page:MoveTo(x, y)`

BESCHREIBUNG

`page:MoveTo()` startet einen neuen Unterpfad und verschiebt den aktuellen Punkt für den Pfad zum Punkt (x, y).

EINGABEN

`x` x Startpunkt für den Pfad
`y` y Startpunkt für den Pfad

RÜCKGABEWERTE

`status` Statusinformation

14.76 page:MoveToNextLine

BEZEICHNUNG

page:MoveToNextLine – verschiebt die aktuelle Position in die nächste Zeile

ÜBERSICHT

```
status = page:MoveToNextLine()
```

BESCHREIBUNG

page:MoveToNextLine() verschiebt die aktuelle Position für den Text in Abhängigkeit vom aktuellen Text, der den Punkt und die Textführung anzeigt. Die neue Position wird mit der aktuellen Textübergangsmatrix berechnet.

Gibt bei Erfolg #HPDF_OK zurück. Andernfalls wird ein Fehlercode zurückgegeben und der Fehler-Handler wird aufgerufen.

EINGABEN

keine

RÜCKGABEWERTE

status Statusinformation

14.77 page:Rectangle

BEZEICHNUNG

page:Rectangle – hängt ein Rechteck an den Pfad an

ÜBERSICHT

```
status = page:Rectangle(x, y, width, height)
```

BESCHREIBUNG

page:Rectangle() hängt ein Rechteck an den aktuellen Pfad an.

EINGABEN

x x-Koordinate des linken unteren Punktes des Rechtecks

y y-Koordinate des linken unteren Punktes des Rechtecks

width Breite des Rechtecks

height Höhe des Rechtecks

RÜCKGABEWERTE

status Statusinformation

14.78 page:SetCharSpace

BEZEICHNUNG

page:SetCharSpace – legt den Zeichenabstand fest

ÜBERSICHT

```
status = page:SetCharSpace(value)
```

BESCHREIBUNG

`page:SetCharSpace()` legt den Zeichenabstand für den Text fest.

EINGABEN

`value` der Zeichenabstand (Anfangswert ist 0)

RÜCKGABEWERTE

`status` Statusinformation

14.79 page:SetCMYKFill**BEZEICHNUNG**

`page:SetCMYKFill` – stellt die CMYK-Füllfarbe ein

ÜBERSICHT

`status = page:SetCMYKFill(c, m, y, k)`

BESCHREIBUNG

`page:SetCMYKFill()` setzt die Füllfarbe. Die einzelnen Parameter müssen alle zwischen 0 und 1 liegen.

EINGABEN

`c` Grünblau-Anteil der Farbe (Cyan)

`m` Magentarot-Anteil der Farbe (violett/magenta)

`y` Gelb-Anteil der Farbe (Yellow)

`k` Schwarz-Anteil der Farbe (Key/Black)

RÜCKGABEWERTE

`status` Statusinformation

14.80 page:SetCMYKStroke**BEZEICHNUNG**

`page:SetCMYKStroke` – stellt die CMYK-Zeichnungsfarbe ein

ÜBERSICHT

`status = page:SetCMYKStroke(c, m, y, k)`

BESCHREIBUNG

`page:SetCMYKStroke()` setzt die Zeichnungsfarbe. Die einzelnen Parameter müssen alle zwischen 0 und 1 liegen.

EINGABEN

`c` Grünblau-Anteil der Farbe (Cyan)

`m` Magentarot-Anteil der Farbe (violett/magenta)

`y` Gelb-Anteil der Farbe (Yellow)

`k` Schwarz-Anteil der Farbe (Key/Black)

RÜCKGABEWERTE

`status` Statusinformation

14.81 `page:SetDash`

BEZEICHNUNG

`page:SetDash` – setzt das Strichmuster für die Linien

ÜBERSICHT

```
status = page:SetDash([pattern, phase])
```

BESCHREIBUNG

`page:SetDash()` setzt das Strichmuster für Linien auf der Seite. Das Muster muss eine Tabelle sein, die zwischen 0 und 8 Elemente von Strichen und Lücken enthält. Wenn Sie ohne Parameter aufgerufen werden, wird das Strichmuster der Linie deaktiviert.

Hier sind einige gängige Muster:

```
page:SetDash({3}, 1)
page:SetDash({7,3}, 2)
page:SetDash({8,7,2,7}, 0)
```

EINGABEN

`pattern` optional: Muster von Strichen und Lücken, die zum Zeichnen von Pfaden verwendet werden.

`phase` optional: die Phase, in der das Muster beginnt (Voreingestellt ist 0).

RÜCKGABEWERTE

`status` Statusinformation

14.82 `page:SetExtGState`

BEZEICHNUNG

`page:SetExtGState` – wendet den erweiterten Grafikstatus an

ÜBERSICHT

```
status = page:SetExtGState(extgstate)
```

BESCHREIBUNG

`page:SetExtGState()` wendet den erweiterten Grafikstatus auf die Seite an.

EINGABEN

`extgstate`
die Kennung eines erweiterten Grafikstatusobjekts

RÜCKGABEWERTE

`status` Statusinformation

14.83 page:SetFlat

BEZEICHNUNG

page:SetFlat – setzt die akute Kurvennäherung

ÜBERSICHT

status = page:SetFlat(flatness)

BESCHREIBUNG

page:SetFlat() setzt den aktuellen Wert der Kurvennäherung der Seite.

EINGABEN

flatness gewünschte Kurvennäherung

RÜCKGABEWERTE

status Statusinformation

14.84 page:SetFontAndSize

BEZEICHNUNG

page:SetFontAndSize – legt die Schriftart und -größe fest

ÜBERSICHT

status = page:SetFontAndSize(font, size)

BESCHREIBUNG

page:SetFontAndSize() setzt die Schriftart und die -größe.

EINGABEN

font die Kennung eines Schriftartobjekts

size die Größe einer Schriftart

RÜCKGABEWERTE

status Statusinformation

14.85 page:SetGrayFill

BEZEICHNUNG

page:SetGrayFill – stellt die graue Füllfarbe ein

ÜBERSICHT

status = page:SetGrayFill(gray)

BESCHREIBUNG

page:SetGrayFill() setzt die graue Füllfarbe.

EINGABEN

value der Wert der Graustufe zwischen 0 und 1

RÜCKGABEWERTE

status Statusinformation

14.86 page:SetGrayStroke

BEZEICHNUNG

page:SetGrayStroke – setzt die graue Zeichnungsfarbe

ÜBERSICHT

```
status = page:SetGrayStroke(gray)
```

BESCHREIBUNG

page:SetGrayStroke() setzt die graue Zeichnungsfarbe.

EINGABEN

value der Wert der Graustufe zwischen 0 und 1

RÜCKGABEWERTE

status Statusinformation

14.87 page:SetHeight

BEZEICHNUNG

page:SetHeight – legt die Seitenhöhe fest

ÜBERSICHT

```
status = page:SetHeight(value)
```

BESCHREIBUNG

page:SetHeight() ändert die Höhe einer Seite.

EINGABEN

value die neue Seitenhöhe; gültige Werte liegen zwischen 3 und 14400

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_PAGE - Es wurde eine ungültige Seiten-Kennung gesetzt.

#HPDF_PAGE_INVALID_SIZE - Es wurde eine ungültige Größe eingestellt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

14.88 page:SetHorizontalScaling

BEZEICHNUNG

page:SetHorizontalScaling – setzt die horizontale Skalierung für den Text

ÜBERSICHT

```
status = page:SetHorizontalScaling(value)
```

BESCHREIBUNG

page:SetHorizontalScaling() setzt die horizontale Skalierung für den Text.

EINGABEN

`value` der Wert der horizontalen Skalierung (anfänglich 100)

RÜCKGABEWERTE

`status` Statusinformation

14.89 page:SetLineCap**BEZEICHNUNG**

`page:SetLineCap` – setzt den aktuellen Stil der Liniendenen

ÜBERSICHT

`status = page:SetLineCap(linecap)`

BESCHREIBUNG

`page:SetLineCap()` setzt die Form, die an den Enden von Linien verwendet werden soll. Der Parameter `linecap` muss eine der folgenden Konstanten sein:

`#HPDF_BUTT_END`

Die Linie wird am Endpunkt des Pfades quadriert.

`#HPDF_ROUND_END`

Das Ende der Linie wird zu einem Halbkreis, dessen Mittelpunkt sich am Bahnendpunkt befindet.

`#HPDF_PROJECTING_SQUARE_END`

Linie fährt über den Endpunkt hinaus fort, geht auf die Hälfte der Endpunkt-Hubbreite.

EINGABEN

`linecap` die gewünschte Linienkappenform (siehe oben)

RÜCKGABEWERTE

`status` Statusinformation

14.90 page:SetLineJoin**BEZEICHNUNG**

`page:SetLineJoin` – setzt den Stil beim Verbinden von Linien

ÜBERSICHT

`status = page:SetLineJoin(linejoin)`

BESCHREIBUNG

`page:SetLineJoin()` legt den Stil für das Verbinden von Linien auf der Seite fest. Der Parameter `linejoin` muss eine der folgenden Konstanten sein:

`#HPDF_MITER_JOIN`

Verwendet eine Gehrungsverbindung (eine scharfkantige Ecke). Dies ist der Standard Verbindungsstil-Modus.

#HPDF_ROUND_JOIN

Verbindet Linien, indem ihre Enden als Kreise gezeichnet wird. Dadurch entsteht ein dicker Stiftabdruck.

#HPDF_BEVEL_JOIN

Verbindet Linien, indem die Linienenden auf der Hälfte der Linienbreite abgeschnitten wird.

EINGABEN

`linejoin` den gewünschten Stil beim Verbinden von Linien (siehe oben)

RÜCKGABEWERTE

`status` Statusinformation

14.91 page:SetLineWidth**BEZEICHNUNG**

`page:SetLineWidth` – setzt die Zeichnungslinienbreite

ÜBERSICHT

`status = page:SetLineWidth(linewidth)`

BESCHREIBUNG

`page:SetLineWidth()` setzt die Breite der Linie, die zum Zeichnen eines Pfades verwendet wird.

EINGABEN

`linewidth`

die zu verwendende Linienbreite (Standard ist 1)

RÜCKGABEWERTE

`status` Statusinformation

14.92 page:SetMiterLimit**BEZEICHNUNG**

`page:SetMiterLimit` – setzt die aktuelle Gehrungsgrenze

ÜBERSICHT

`status = page:SetMiterLimit(miterlimit)`

BESCHREIBUNG

Legt die Gehrungsgrenze fest. Die Voreinstellung ist 10.

EINGABEN

`miterlimit`

gewünschte Gehrungsgrenze

RÜCKGABEWERTE

`status` Statusinformation

14.93 page:SetRGBFill

BEZEICHNUNG

page:SetRGBFill – setzt die aktuelle RGB-Füllfarbe

ÜBERSICHT

```
status = page:SetRGBFill(r, g, b)
```

BESCHREIBUNG

page:SetRGBFill() setzt die RGB-Füllfarbe. Die einzelnen Farbkomponenten müssen zwischen 0 und 1 liegen.

EINGABEN

r Rot-Wert der neuen Farbe
g Grün-Wert der neuen Farbe
b Blau-Wert der neuen Farbe

RÜCKGABEWERTE

status Statusinformation

14.94 page:SetRGBStroke

BEZEICHNUNG

page:SetRGBStroke – setzt die RGB-Zeichnungsfarbe

ÜBERSICHT

```
status = page:SetRGBStroke(r, g, b)
```

BESCHREIBUNG

page:SetRGBStroke() setzt die Zeichnungsfarbe. Die einzelnen Farbkomponenten müssen zwischen 0 und 1 liegen.

EINGABEN

r Rot-Wert der neuen Farbe
g Grün-Wert der neuen Farbe
b Blau-Wert der neuen Farbe

RÜCKGABEWERTE

status Statusinformation

14.95 page:SetRotate

BEZEICHNUNG

page:SetRotate – stellt die Seitendrehung ein

ÜBERSICHT

```
status = page:SetRotate(angle)
```

BESCHREIBUNG

`page:SetRotate()` setzt den Drehwinkel der Seite.

EINGABEN

`angle` der Drehwinkel der Seite; er muss ein Vielfaches von 90 Grad betragen.

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

#HPDF_INVALID_PAGE - Es wurde eine ungültige Seiten-Kennung gesetzt.

#HPDF_PAGE_INVALID_ROTATE_VALUE - Ein ungültiger Drehwinkel wurde eingestellt.

14.96 page:SetSize

BEZEICHNUNG

`page:SetSize` – setzt die Seitengröße und -richtung fest

ÜBERSICHT

`status = page:SetSize(size, direction)`

BESCHREIBUNG

`page:SetSize()` ändert die Größe und Richtung einer Seite auf eine vordefinierte Größe.

Der Parameter `size` muss eine der folgenden Konstanten sein:

#HPDF_PAGE_SIZE_LETTER

8,5 x 11 Zoll (612 x 792 Pixel)

#HPDF_PAGE_SIZE_LEGAL

8.5 x 14 Zoll (612 x 1008 Pixel)

#HPDF_PAGE_SIZE_A3

297 x 420mm (841.89 x 1199.551 Pixel)

#HPDF_PAGE_SIZE_A4

210 x 297mm (595.276 x 841.89 Pixel)

#HPDF_PAGE_SIZE_A5

148 x 210mm (419.528 x 595.276 Pixel)

#HPDF_PAGE_SIZE_B4

250 x 353mm (708.661 x 1000.63 Pixel)

#HPDF_PAGE_SIZE_B5

176 x 250mm (498.898 x 708.661 Pixel)

#HPDF_PAGE_SIZE_EXECUTIVE

7.25 x 10.5 Zoll (522 x 756 Pixel)

#HPDF_PAGE_SIZE_US4x6

4 x 6 Zoll (288 x 432 Pixel)

#HPDF_PAGE_SIZE_US4x8

4 x 8 Zoll (288 x 576 Pixel)

#HPDF_PAGE_SIZE_US5x7
5 x 7 Zoll (360 x 504 Pixel)

#HPDF_PAGE_SIZE_COMM10
4.125 x 9.5 Zoll (297 x 684 Pixel)

Der Parameter `direction` muss eine der folgenden Konstanten sein:

#HPDF_PAGE_PORTRAIT
Stellt den größeren Wert auf Vertikal (Hochformat).

#HPDF_PAGE_LANDSCAPE
Stellt den größeren Wert auf Horizontal (Querformat).

EINGABEN

`size` vordefinierter Seitengrößenwert (siehe oben)

`direction`
die Richtung der Seite (siehe oben für mögliche Werte)

RÜCKGABEWERTE

`status` Statusinformation

FEHLER

#HPDF_INVALID_PAGE - Es wurde eine ungültige Seiten-Kennung gesetzt.

#HPDF_PAGE_INVALID_SIZE - Es wurde eine ungültige Größe eingestellt.

#HPDF_PAGE_INVALID_DIRECTION - Eine ungültige Richtung wurde eingestellt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

14.97 page:SetSlideShow

BEZEICHNUNG

`page:SetSlideShow` – setzt den Seitenwechselmodus

ÜBERSICHT

`status = page:SetSlideShow(type, disptime, transtime)`

BESCHREIBUNG

`page:SetSlideShow()` konfiguriert die Einstellung für den Folienwechsel der Seite. Die `disptime` gibt die Anzeigedauer der Seite in Sekunden an, während der Parameter `transtime` auf die Dauer des Übergangseffekts in Sekunden eingestellt werden muss.

Der Parameter `type` konfiguriert den eigentlichen Effekt und kann einer der folgenden Werte sein:

```
#HPDF_TS_WIPE_RIGHT
#HPDF_TS_WIPE_UP
#HPDF_TS_WIPE_LEFT
#HPDF_TS_WIPE_DOWN
#HPDF_TS_BARN_DOORS_HORIZONTAL_OUT
#HPDF_TS_BARN_DOORS_HORIZONTAL_IN
#HPDF_TS_BARN_DOORS_VERTICAL_OUT
```



```
#HPDF_TS_BARN_DOORS_VERTICAL_IN
#HPDF_TS_BOX_OUT
#HPDF_TS_BOX_IN
#HPDF_TS_BLINDS_HORIZONTAL
#HPDF_TS_BLINDS_VERTICAL
#HPDF_TS DISSOLVE
#HPDF_TS_GLITTER_RIGHT
#HPDF_TS_GLITTER_DOWN
#HPDF_TS_GLITTER_TOP_LEFT_TO_BOTTOM_RIGHT
#HPDF_TS_REPLACE
```

EINGABEN

`type` der Übergangsstil (siehe oben für mögliche Werte)
`disptime` die Anzeigedauer der Seite (in Sekunden)
`transtime` die Dauer des Übergangseffekts (in Sekunden)

RÜCKGABEWERTE

`status` Statusinformation

14.98 page:SetTextLeading**BEZEICHNUNG**

`page:SetTextLeading` – setzt den Zeilenabstand

ÜBERSICHT

`status = page:SetTextLeading(value)`

BESCHREIBUNG

`page:SetTextLeading()` setzt den Zeilenabstand für die Anzeige von Text.

EINGABEN

`value` der Wert für den Zeilenabstand (Initialwert ist 0)

RÜCKGABEWERTE

`status` Statusinformation

14.99 page:SetTextMatrix**BEZEICHNUNG**

`page:SetTextMatrix` – setzt die Texttransformationsmatrix

ÜBERSICHT

`status = page:SetTextMatrix(a, b, c, d, x, y)`

BESCHREIBUNG

`page:SetTextMatrix()` setzt eine Transformationsmatrix für den Text, der mit `page:ShowText()` gezeichnet werden soll. Der Befehl `page:TextRect()` verwendet nicht die aktive Textmatrix.

Liefert bei Erfolg #HPDF_OK, ansonsten einen Fehlercode.

EINGABEN

- a Skalierung der x-Koordinate (Größenveränderung)
- b Rotation der x-Koordinate (Drehung)
- c Rotation der y-Koordinate (Drehung)
- d Skalierung der y-Koordinate (Größenveränderung)
- x Translation der x-Koordinate (Verschiebung)
- y Translation der y-Koordinate (Verschiebung)

RÜCKGABEWERTE

status Statusinformation

14.100 page:SetTextRenderingMode

BEZEICHNUNG

page:SetTextRenderingMode – stellt den Textwiedergabemodus ein

ÜBERSICHT

status = page:SetTextRenderingMode(mode)

BESCHREIBUNG

page:SetTextRenderingMode() setzt den Textwiedergabemodus.

Der Parameter mode muss eine der folgenden Konstanten sein:

```
#HPDF_FILL
#HPDF_STROKE
#HPDF_FILL_THEN_STROKE
#HPDF_INVISIBLE
#HPDF_FILL_CLIPPING
#HPDF_STROKE_CLIPPING
#HPDF_FILL_STROKE_CLIPPING
#HPDF_CLIPPING
```

Der Standard-Textwiedergabemodus ist #HPDF_FILL.

EINGABEN

mode der Textwiedergabemodus (siehe oben für mögliche Modi)

RÜCKGABEWERTE

status Statusinformation

14.101 page:SetTextRise

BEZEICHNUNG

page:SetTextRise – moduliert die y-Position des Textes

ÜBERSICHT

```
status = page:SetTextRise(value)
```

BESCHREIBUNG

page:SetTextRise() verschiebt die Textlage in vertikaler Richtung um den Betrag von value. Nützlich für die Erstellung von hoch- oder tiefgestellten Texten (Indizes).

EINGABEN

value Textlage, in Benutzerabstandseinheiten

RÜCKGABEWERTE

status Statusinformation

14.102 page:SetWidth

BEZEICHNUNG

page:SetWidth – setzt die Seitenbreite

ÜBERSICHT

```
status = page:SetWidth(value)
```

BESCHREIBUNG

page:SetWidth() wechselt die Breite der Seite.

EINGABEN

value die neue Seitenbreite; gültige Werte liegen zwischen 3 und 14400

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_PAGE - Es wurde eine ungültige Seiten-Kennung gesetzt.

#HPDF_PAGE_INVALID_SIZE - Es wurde eine ungültige Größe eingestellt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

14.103 page:SetWordSpace

BEZEICHNUNG

page:SetWordSpace – setzt den Wortabstand

ÜBERSICHT

```
status = page:SetWordSpace(value)
```

BESCHREIBUNG

page:SetWordSpace() setzt den Wortabstand für den Text.

EINGABEN

`value` der Wert des Wortabstandes (Anfangswert ist 0)

RÜCKGABEWERTE

`status` Statusinformation

14.104 page:SetZoom**BEZEICHNUNG**

`page:SetZoom` – stellt den Seitenzoom ein

ÜBERSICHT

`status = page:SetZoom(zoom)`

BESCHREIBUNG

`page:SetZoom()` setzt den Zoomfaktor für die Seite.

EINGABEN

`zoom` die gewünschte Zoomeinstellung

RÜCKGABEWERTE

`status` Statusinformation

14.105 page:ShowText**BEZEICHNUNG**

`page:ShowText` – gibt den Text aus

ÜBERSICHT

`status = page:ShowText(text)`

BESCHREIBUNG

`page:ShowText()` gibt den Text an der aktuellen Position auf der Seite aus.

EINGABEN

`text` der Text, der ausgegeben wird

RÜCKGABEWERTE

`status` Statusinformation

14.106 page:ShowTextNextLine**BEZEICHNUNG**

`page:ShowTextNextLine` – bricht die Zeile um und gibt den Text aus

ÜBERSICHT

`status = page:ShowTextNextLine(text[, wordspace, charspace])`

BESCHREIBUNG

`page:ShowTextNextLine()` verschiebt die aktuelle Textposition an den Anfang der nächsten Zeile und gibt den Text dann an der aktuellen Position auf der Seite aus. Wenn die optionalen Argumente `wordspace` und `charspace` angegeben sind, setzt diese Methode auch den Wort- und Zeichenabstand vor dem Ausgeben des Textes.

EINGABEN

`text` der Text, der ausgegeben wird

`wordspace`
 optional: Wortabstand für den Text

`charspace`
 optional: Zeichenabstand für den Text

RÜCKGABEWERTE

`status` Statusinformation

14.107 page:Stroke**BEZEICHNUNG**

`page:Stroke` – zeichnet den aktuellen Pfad

ÜBERSICHT

`status = page:Stroke()`

BESCHREIBUNG

`page:Stroke()` zeichnet den aktuellen Pfad.

EINGABEN

keine

RÜCKGABEWERTE

`status` Statusinformation

14.108 page:TextOut**BEZEICHNUNG**

`page:TextOut` – gibt den Text an der angegebenen Position aus

ÜBERSICHT

`status = page:TextOut(xpos, ypos, text)`

BESCHREIBUNG

`page:TextOut()` gibt den Text an der angegebenen Position aus.

EINGABEN

`xpos` x-Position, an der der Text ausgegeben werden soll

`ypos` y-Position, an der der Text ausgegeben werden soll

`text` der Text, der ausgegeben werden soll

RÜCKGABEWERTE

`status` Statusinformation

14.109 `page:TextRect`

BEZEICHNUNG

`page:TextRect` – gibt den Text im Bereich aus

ÜBERSICHT

`status, len = page:TextRect(left, top, right, bottom, text, align)`

BESCHREIBUNG

`page:TextRect()` gibt den Text innerhalb des angegebenen Bereichs aus.

Der Parameter `align` muss eine der folgenden Konstanten sein:

`#HPDF_TALIGN_LEFT`

Der Text ist links ausgerichtet.

`#HPDF_TALIGN_RIGHT`

Der Text ist rechts ausgerichtet.

`#HPDF_TALIGN_CENTER`

Der Text ist zentriert.

`#HPDF_TALIGN_JUSTIFY`

Fügt Leerzeichen zwischen den Wörtern hinzu, um sowohl links als auch rechts ausgerichtet zu sein (Blocksatz).

EINGABEN

`left` linke Koordinate des Bereichs

`top` obere Koordinate des Bereichs

`rechts` rechte Koordinate des Bereichs

`untere` untere Koordinate des Bereichs

`text` der Text, der ausgegeben werden soll

`align` die Ausrichtung des Textes (eine der oben genannten Optionen)

RÜCKGABEWERTE

`status` Statusinformation

`len` die Anzahl der im Bereich ausgegebenen Zeichen

14.110 `page:TextWidth`

BEZEICHNUNG

`page:TextWidth` – ermittelt die Textbreite

ÜBERSICHT

`w = page:TextWidth(text)`

BESCHREIBUNG

`page:TextWidth()` ermittelt die Breite des Textes in der aktuellen Schriftgröße, Zeichenabstand und Wortabstand.

EINGABEN

`text` der Text, dessen Breite ermittelt werden soll

RÜCKGABEWERTE

`w` Textbreite

15 Umrissmethoden

15.1 outline:SetDestination

BEZEICHNUNG

outline:SetDestination – setzt das Zielobjekt

ÜBERSICHT

status = outline:SetDestination(dst)

BESCHREIBUNG

outline:SetDestination() setzt ein Zielobjekt, zu dem man springen kann, wenn man auf die Gliederung klickt.

EINGABEN

dst die Kennung eines Zielobjekts

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_OUTLINE - Eine ungültige Gliederungs-Kennung wurde gesetzt.

#HPDF_INVALID_DESTINATION - Es wurde eine ungültige Ziel-Kennung gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

15.2 outline:SetOpened

BEZEICHNUNG

outline:SetOpened – setzt den Öffnungsmodus des Knotens

ÜBERSICHT

status = outline:SetOpened(opened)

BESCHREIBUNG

outline:SetOpened() legt fest, ob dieser Knoten beim ersten Anzeigen der Gliederung geöffnet wird oder nicht.

EINGABEN

opened gibt an, ob der Knoten geöffnet wird oder nicht

RÜCKGABEWERTE

status Statusinformation

FEHLER

#HPDF_INVALID_OUTLINE - Eine ungültige Gliederungs-Kennung wurde gesetzt.

#HPDF_FAILED_TO_ALLOC_MEM - Speicherzuordnung ist fehlgeschlagen.

Anhang A Lizenzen

A.1 LibHaru license

Copyright (C) 1999-2006 Takeshi Kanno

Copyright (C) 2007-2009 Antony Dovgal

This software is provided 'as-is', without any express or implied warranty.

In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

A.2 LuaHPDF license

LuaHPDF is Copyright 2007-2013 by Kurt Jung.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A.3 PDFium license

Copyright 2014 PDFium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Index

A

annot:SetBorderStyle	43
annot:SetCMYKColor	43
annot:SetFreeTextAnnot2PointCalloutLine	44
annot:SetFreeTextAnnot3PointCalloutLine	45
annot:SetFreeTextAnnotDefaultStyle	45
annot:SetFreeTextAnnotLineEndingStyle	46
annot:SetGrayColor	47
annot:SetLineAnnotCaption	47
annot:SetLineAnnotLeader	48
annot:SetLineAnnotPosition	49
annot:SetLinkAnnotBorderStyle	49
annot:SetLinkAnnotHighlightMode	50
annot:SetMarkupAnnotCloudEffect	51
annot:SetMarkupAnnotCreationDate	51
annot:SetMarkupAnnotIntent	52
annot:SetMarkupAnnotInteriorCMYKColor	53
annot:SetMarkupAnnotInteriorGrayColor	53
annot:SetMarkupAnnotInteriorRGBColor	54
annot:SetMarkupAnnotInteriorTransparent	54
annot:SetMarkupAnnotPopup	55
annot:SetMarkupAnnotQuadPoints	55
annot:SetMarkupAnnotRectDiff	56
annot:SetMarkupAnnotSubject	56
annot:SetMarkupAnnotTitle	57
annot:SetMarkupAnnotTransparency	57
annot:SetNoColor	57
annot:SetPopupAnnotOpened	58
annot:SetRGBColor	58
annot:SetTextAnnotIcon	59
annot:SetTextAnnotOpened	60

D

dest:SetFit	61
dest:SetFitB	61
dest:SetFitBH	61
dest:SetFitBV	62
dest:SetFitH	62
dest:SetFitR	63
dest:SetFitV	63
dest:SetXYZ	64
doc:AddPage	69
doc:AddPageLabel	69
doc:AttachFile	70
doc:CreateExtGState	70
doc:CreateImageFromBrush	71
doc:CreateImageFromMem	72
doc:CreateOutline	73
doc:Free	73
doc:GetCurrentEncoder	74
doc:GetCurrentPage	74
doc:GetEncoder	74
doc:GetError	75

doc:GetErrorDetail	75
doc:GetFont	76
doc:GetInfoAttr	77
doc:GetPageByIndex	77
doc:GetPageLayout	78
doc:GetPageMode	78
doc:GetViewerPreference	78
doc:InsertPage	79
doc:LoadFont	79
doc:LoadJPEGImage	81
doc:LoadPNGImage	81
doc:LoadRawImage	82
doc:LoadTTFont	83
doc:LoadType1Font	84
doc:ResetError	85
doc:SaveToFile	85
doc:SetCompressionMode	86
doc:SetCurrentEncoder	87
doc:SetEncryptionMode	87
doc:SetInfoAttr	88
doc:SetInfoDateAttr	89
doc:SetOpenAction	90
doc:SetPageLayout	90
doc:SetPageMode	91
doc:SetPagesConfiguration	92
doc:SetPassword	92
doc:SetPermission	93
doc:SetViewerPreference	94
doc:UseCNSEncodings	94
doc:UseCNSFonts	95
doc:UseCNTEncodings	96
doc:UseCNTFonts	96
doc:UseJPEncodings	97
doc:UseJPFonts	98
doc:UseKREncodings	99
doc:UseKRFonts	99
doc:UseUTFEncodings	100

E

encoder:GetByteType	105
encoder:GetType	105
encoder:GetUnicode	105
encoder:GetWritingMode	106
extgs:SetAlphaFill	103
extgs:SetAlphaStroke	103
extgs:SetBlendMode	104

F

font:GetAscent	107
font:GetBBox	107
font:GetCapHeight	107
font:GetDescent	108
font:GetEncodingName	108
font:GetFontName	108
font:GetUnicodeWidth	109
font:GetXHeight	109
font:MeasureText	110
font:TextWidth	110

I

image:AddSMask	65
image:GetBitsPerComponent	65
image:GetColorSpace()	65
image:GetHeight	66
image:GetSize	66
image:GetWidth	66
image:SetColorMask	67
image:SetMaskImage	67

O

outline:SetDestination	167
outline:SetOpened	167

P

page:Arc	113
page:BeginText	113
page:Circle	113
page:Clip	114
page:ClosePath	114
page:ClosePathEofillStroke	115
page:ClosePathFillStroke	115
page:ClosePathStroke	116
page:Concat	116
page>CreateCircleAnnot	117
page>CreateDestination	118
page>CreateFreeTextAnnot	118
page>CreateHighlightAnnot	119
page>CreateLineAnnot	119
page>CreateLinkAnnot	120
page>CreatePopupAnnot	120
page>CreateProjectionAnnot	121
page>CreateSquareAnnot	121
page>CreateSquigglyAnnot	122
page>CreateStampAnnot	123
page>CreateStrikeOutAnnot	124
page>CreateTextAnnot	124
page>CreateTextMarkupAnnot	125
page>CreateUnderlineAnnot	126
page>CreateURILinkAnnot	126
page>CreateWidgetAnnot	127
page:CurveTo	127
page:CurveTo2	128

page:CurveTo3	128
page:DrawImage	129
page:Ellipse	129
page:EndPath	130
page:EndText	130
page:EoClip	131
page:Eofill	131
page:EofillStroke	131
page:ExecuteXObject	132
page:Fill	132
page:FillStroke	132
page:GetCharSpace	133
page:GetCMYKFill	133
page:GetCMYKStroke	134
page:GetCurrentFont	134
page:GetCurrentFontSize	135
page:GetCurrentPos	135
page:GetCurrentTextPos	135
page:GetDash	136
page:GetFillingColorSpace	136
page:GetFlat	137
page:GetGMode	137
page:GetGrayFill	138
page:GetGrayStroke	138
page:GetGStateDepth	138
page:GetHeight	139
page:GetHorizontalScaling	139
page:GetLineCap	139
page:GetLineJoin	140
page:GetLineWidth	140
page:GetMiterLimit	141
page:GetRGBFill	141
page:GetRGBStroke	141
page:GetStrokingColorSpace	142
page:GetTextLeading	142
page:GetTextMatrix	143
page:GetTextRenderingMode	143
page:GetTextRise	144
page:GetTransMatrix	144
page:GetWidth	145
page:GetWordSpace	145
page:GRestore	145
page:GSave	146
page:LineTo	147
page:MeasureText	147
page:MoveTextPos	148
page:MoveTo	148
page:MoveToNextLine	148
page:Rectangle	149
page:SetCharSpace	149
page:SetCMYKFill	150
page:SetCMYKStroke	150
page:SetDash	151
page:SetExtGState	151
page:SetFlat	151
page:SetFontAndSize	152
page:SetGrayFill	152
page:SetGrayStroke	152

page:SetHeight	153	pdf.FindNext	22
page:SetHorizontalScaling	153	pdf.FindPrev	23
page:SetLineCap	154	pdf.FindStart	24
page:SetLineJoin	154	pdf.FreePage	25
page:SetLineWidth	155	pdf.GetBookmarks	25
page:SetMiterLimit	155	pdf.GetBoundedText	26
page:SetRGBFill	155	pdf.GetBrush	27
page:SetRGBStroke	156	pdf.GetBrushFromPage	28
page:SetRotate	156	pdf.GetCharBox	29
page:SetSize	157	pdf.GetCharIndexAtPos	30
page:SetSlideShow	158	pdf.GetCharOrigin	30
page:SetTextLeading	159	pdf.GetCropBox	31
page:SetTextMatrix	159	pdf.GetFindResult	31
page:SetTextRenderingMode	160	pdf.GetLastError	32
page:SetTextRise	160	pdf.GetMediaBox	33
page:SetWidth	161	pdf.GetMetaText	33
page:SetWordSpace	161	pdf.GetObjectType	34
page:SetZoom	162	pdf.GetPageLabel	35
page:ShowText	162	pdf.GetPageLen	36
page:ShowTextNextLine	162	pdf.GetPageLinks	36
page:Stroke	163	pdf.GetRects	37
page:TextOut	163	pdf.GetText	38
page:TextRect	164	pdf.GetVersion	39
page:TextWidth	164	pdf.IsPDF	39
pdf.CloseDocument	21	pdf.LoadPage	39
pdf.CreateDocument	21	pdf.OpenDocument	40
pdf.DeviceToPage	21	pdf.PageToDevice	41

