

# 6 Variable Daten und Blöcke

PDFlib bietet einen PDF-Workflow zur Verarbeitung variabler Daten auf Basis von Templates. Mit Hilfe des Blockkonzepts können importierte Seiten mit variablen Mengen von ein- oder mehrzeiligem Text, Rasterbildern oder PDF-Vektorgrafik angereichert werden, die aus externen Quellen bezogen werden. Damit lassen sich problemlos Anwendungen implementieren, die individualisierte PDF-Dokumente erfordern, zum Beispiel:

- ▶ Serienbriefe
- ▶ flexible Erstellung personalisierter Massenbriefe
- ▶ Berichtswesen und Rechnungserstellung
- ▶ Personalisierung von Visitenkarten

*Hinweis* Zur Blockverarbeitung ist der PDFlib Personalization Server (PPS) erforderlich. PPS ist zwar in allen kommerziellen PDFlib-Paketen enthalten, Sie müssen dafür aber einen eigenen Lizenzschlüssel erwerben; ein Lizenzschlüssel für PDFlib oder PDFlib+PDI reicht nicht aus. Außerdem ist das Block-Plugin für Adobe Acrobat erforderlich, um Blöcke in PDF-Templates interaktiv anzulegen.

## 6.1 Installation des Block-Plugins

Das Block-Plugin sowie das Plugin zur Konvertierung von Formularfeldern funktionieren nur mit der Vollversion von Acrobat 5, mit Acrobat 6/7 Standard und Acrobat 6/7 Professional unter Windows und auf dem Mac. Mit Acrobat 6/7 Elements und Acrobat Reader/Adobe Reader sind sie nicht einsetzbar.

*Hinweis* Das Plugin enthält die Benutzeroberfläche in verschiedenen Sprachversionen und verwendet, soweit möglich, dieselbe Sprache wie Acrobat selbst. Wird diese Sprache vom Plugin nicht unterstützt, verwendet es stattdessen Englisch.

**Installation der Block-Plugins für Acrobat 5/6/7 unter Windows.** Zur Installation des Block-Plugins sowie des Plugins zur Konvertierung von PDF-Formularfeldern in Acrobat 5, 6 oder 7 kopieren Sie die Plugin-Dateien in ein Unterverzeichnis des Acrobat-Plugin-Verzeichnisses. Dies wird von der Installationsroutine des Plugins automatisch durchgeführt, kann aber auch manuell erfolgen. Unter Windows heißen die Dateien *Block.api* und *AcroFormConversion.api*. Das Verzeichnis für die PDFlib-Plugins lautet üblicherweise in etwa:

```
C:\Programme\Adobe\Acrobat 7.0\Acrobat\plug_ins\PDFlib Block Plugin
```

**Installation der Block-Plugins für Acrobat 6/7 auf dem Mac.** Bei Acrobat 6/7 ist der Plugin-Ordner im Finder nicht sichtbar. Statt die Plugin-Dateien in den Plugin-Ordner zu ziehen, gehen Sie wie folgt vor (beenden Sie Acrobat, falls geöffnet):

- ▶ Durch Doppelklick auf das Laufwerkssymbol extrahieren Sie die Plugin-Dateien in einen Ordner.
- ▶ Suchen Sie im Finder nach dem Symbol für die Acrobat-Anwendung. Normalerweise befindet es sich in einem Verzeichnis, das in etwa folgendermaßen lautet:

```
/Programme/Adobe Acrobat 7.0 Professional
```

- ▶ Nach einem Einfachklick auf das Symbol der Acrobat-Anwendung selektieren Sie *Datei, Info*. Ein Fenster erscheint.
- ▶ Klicken Sie auf das Dreieck neben *Plug-ins*.
- ▶ Klicken Sie auf *Hinzufügen...* und selektieren Sie den Ordner *PDFlib Block Plugin* (in Acrobat 7) bzw. den Ordner *PDFlib Block Plugin Acro 5-6* (in Acrobat 6), der sich in dem Ordner befindet, der im ersten Schritt angelegt wurde. Beachten Sie, dass dieser Ordner nach der Installation nicht sofort in der Plugin-Liste erscheint. Er wird erst beim nächsten Öffnen des Info-Fensters angezeigt.

**Installation der Block-Plugins für Acrobat 5 auf dem Mac.** Zur Installation der Plugins für Acrobat 5 doppelklicken Sie auf das Laufwerkssymbol. Ziehen Sie den Ordner *PDFlib Block Plugin Acro 5-6* auf den Plugin-Ordner von Acrobat 5, der üblicherweise wie folgt lautet:

/Programme/Acrobat 5.0/Plug-Ins

**Fehlerbehebung.** Falls das Block-Plugin nicht zu funktionieren scheint, stellen Sie sicher, dass unter *Bearbeiten, Grundeinstellungen...*, [*Allgemein...*], *Programmstart* (Acrobat 6/7) bzw. *Bearbeiten, Grundeinstellungen, Allgemein...*, *Optionen* (Acrobat 5) das Kontrollkästchen »Nur zertifizierte Zusatzmodule verwenden« deaktiviert ist. Die Plugins funktionieren sonst nicht.

## 6.2 Überblick über das Blockkonzept von PDFlib

### 6.2.1 Vollständige Trennung von Dokumentdesign und Programmcode

Mit PDFlib-Datenblöcken ist es problemlos möglich, variablen Text, Vektorgrafik oder Rasterbilder über importierte Seiten zu platzieren. Im Gegensatz zu einfachen PDF-Seiten enthalten Seiten mit Datenblöcken Informationen über die Art der Verarbeitung, die später auf der Serverseite stattfindet. Das Blockkonzept von PDFlib trennt die folgenden Aufgaben vollständig voneinander:

- ▶ Ein Designer erstellt das Seitenlayout und legt dabei die Position von variablem Text und Bildelementen sowie deren Eigenschaften wie Schriftgröße, Farbe oder Bildskalierung fest. Nach der Erstellung des Layouts als PDF-Dokument legt der Designer mit dem Block-Plugin für Acrobat die variablen Datenblöcke und ihre Eigenschaften fest.
- ▶ Ein Programmierer schreibt den Code, der die Informationen in den PDF-Blöcken auf den importierten PDF-Seiten mit dynamischen Daten wie zum Beispiel Datenbankfeldern verknüpft. Der Programmierer benötigt keine genaueren Kenntnisse über einen Block (ob dieser einen Namen oder eine Postleitzahl enthält, wo genau er sich auf der Seite befindet oder wie er formatiert ist etc.). Er ist deshalb von Layoutänderungen unabhängig. PDFlib kümmert sich um alle blockspezifischen Details, die aus den in der Datei abgelegten Blockeigenschaften ermittelt werden.

Anders ausgedrückt ist der vom Programmierer entwickelte Code »datenblind«, das heißt, er ist generisch und hängt nicht von blockspezifischen Eigenschaften ab. Angenommen, der Designer beschließt, in einem Serienbrief statt des Nachnamens den Vornamen des Empfängers zu verwenden. Der generische Code zur Blockverarbeitung braucht dann nicht geändert zu werden. Die Ausgabe wird korrekt generiert, sobald der Designer die Blockeigenschaften mit dem Acrobat-Plugin entsprechend geändert hat.

**Beispiel: Hinzufügen von variablem Text zu einem Template.** Eine häufige Aufgabe besteht darin, ein PDF-Template mit dynamischem Text anzureichern. Das folgende Codefragment öffnet eine Seite in einem Eingabe-Dokument (dem PDF-Template), platziert diese auf der Ausgabeseite und füllt einen Textblock namens *firstname* mit variablem Text:

```
doc = PDF_open_pdi(p, filename, "", 0);
if (doc == -1) {
    printf("PDF-Template '%s' konnte nicht geöffnet werden\n", filename);
    return (1);
}
page = PDF_open_pdi_page(p, doc, pageno, "");
if (page == -1) {
    printf("Seite %d von PDF-Template '%s' konnte nicht geöffnet werden\n",
        pageno, filename);
    return (2);
}

PDF_begin_page_ext(p, width, height, "");
PDF_fit_pdi_page(p, page, 0.0, 0.0, "");
PDF_fill_textblock(p, page, "firstname", "Serge", 0, "encoding winansi");
PDF_close_pdi_page(p, page);
PDF_end_page_ext(p, "");
```

## 6.2.2 Blockeigenschaften

Das Verhalten von Blöcken lässt sich über die Blockeigenschaften steuern. Diese werden einem Block mit dem Block-Plugin für Acrobat zugeordnet.

**Standardblockeigenschaften.** PDFlib-Blöcke sind als Rechtecke auf der Seite definiert, denen ein Name, ein Typ und eine offene Menge von Eigenschaften zugewiesen sind, die später auf der Serverseite verarbeitet werden. Der Name entspricht einem beliebigen String zur Identifikation des Blocks, zum Beispiel *firstname*, *lastname* oder *zipcode*. PDFlib unterstützt folgende Blocktypen:

- ▶ Blöcke vom Typ *Text* enthalten eine oder mehrere Zeilen mit Textdaten. Mehrzeiliger Text wird mit der Textflussfunktion formatiert. Beachten Sie, dass sich Blöcke nicht so verbinden lassen, dass Text von einem Block in den nächsten fließt.
- ▶ Blöcke vom Typ *Image* enthalten ein Rasterbild. Dies entspricht im Wesentlichen dem Import eines TIFF- oder JPEG-Bildes in einer DTP-Anwendung.
- ▶ Blöcke vom Typ *PDF* enthalten beliebige PDF-Grafik, die aus einer Seite eines anderen PDF-Dokuments importiert wurde. Dies entspricht im Wesentlichen dem Import einer EPS-Grafik in einer DTP-Anwendung.

Ein Block verfügt abhängig vom jeweiligen Typ über bestimmte Standardeigenschaften. So kann für einen Textblock zum Beispiel die Schriftart und -größe des Texts und für einen Bild- oder PDF-Block der Skalierungsfaktor oder die Drehung festgelegt werden. Für jeden Blocktyp bietet das PDFlib-API eine eigene Verarbeitungsfunktion. Anhand des Blocknamens suchen die Funktionen in einer importierten PDF-Seite nach dem Block, analysieren seine Eigenschaften und platzieren die vom Client übergebenen Daten (Text, Rasterbild oder PDF-Seite) entsprechend der jeweiligen Blockeigenschaften auf der neuen Seite.

**Selbstdefinierte Blockeigenschaften.** Die standardmäßig vorhandenen Blockeigenschaften ermöglichen eine schnelle Implementierung von Anwendungen zur Verarbeitung variabler Daten. Designer sind damit jedoch auf die Eigenschaften beschränkt, die PDFlib intern kennt und automatisch verarbeitet. Um mehr Flexibilität zu bieten, wird deshalb zusätzlich die Möglichkeit geboten, einen Block mit selbstdefinierten Eigenschaften zu versehen. Durch diese Erweiterung wird das Blockkonzept selbst Anwendungen gerecht, die höchste Anforderungen an die Verarbeitung variabler Daten stellen.

Für selbstdefinierte Eigenschaften gibt es keinerlei Regeln, da sie von PDFlib in keiner Weise verarbeitet, sondern lediglich dem Client verfügbar gemacht werden, der sie inspizieren und auf geeignete Art darauf reagieren kann. Auf Basis der selbstdefinierten Eigenschaften eines Blocks könnte zum Beispiel über Layout oder Datenerfassung entschieden werden. So könnte für eine wissenschaftliche Anwendung festgelegt werden, mit wie vielen Stellen eine Zahl ausgegeben wird, oder die Blockeigenschaft könnte den Namen eines Datenbankfeldes definieren, dessen Inhalt dann zum Füllen des Blocks benutzt wird.

**Überschreiben von Blockeigenschaften.** In manchen Situationen möchte der Programmierer nur gewisse Eigenschaften in einer Blockdefinition verwenden, andere dagegen mit eigenen Werten überschreiben. Dies kann in verschiedenen Situationen nützlich sein:

- ▶ Der Skalierungsfaktor für ein Bild oder eine PDF-Seite wird nicht der Blockdefinition entnommen, sondern berechnet.
- ▶ Die Blockkoordinaten werden vom Programm angepasst, etwa um eine Rechnung mit einer variablen Zahl von Einträgen zu erstellen.
- ▶ Im Programm werden individuelle Schmuckfarbnamen angegeben, um bei der Erstellung von Drucksachen den Anforderungen einzelner Kunden gerecht zu werden.

Blockeigenschaften können überschrieben werden, indem man den Namen der Blockeigenschaft und die gewünschten Werte in der Optionsliste der `PDF_fill_*block()`-Funktionen wie folgt angibt:

```
PDF_fill_textblock(p, page, "firstname", "Serge", 0, "fontsize 12");
```

Diese Anweisung überschreibt die interne Eigenschaft `fontsize` des Blocks mit dem angegebenen Wert 12. Fast alle Namen allgemeiner Blockeigenschaften können als Optionen benutzt werden; außerdem die Eigenschaften für den jeweiligen Blocktyp. So ist zum Beispiel die Option `underline` nur bei `PDF_fill_textblock()` erlaubt, während die Option `scale` sowohl bei `PDF_fill_imageblock()` als auch bei `PDF_fill_pdfblock()` erlaubt ist, da `scale` sowohl für Image- als auch PDF-Blöcke eine gültige Blockeigenschaft ist.

Das Überschreiben von Blockeigenschaften wirkt sich nur auf den jeweiligen Funktionsaufruf aus. Die Änderung wird nicht in der Blockdefinition gespeichert.

**Koordinatensysteme.** Die Koordinaten für einen Block beziehen sich auf das Standardkoordinatensystem von PDF. Wird die Seite, die den Block enthält, auf der Ausgabeseite platziert, können an `PDF_fit_pdi_page()` verschiedene Optionen zur Positionierung und Skalierung übergeben werden. Diese Parameter werden bei der Verarbeitung des Blocks berücksichtigt. Damit wird es möglich, eine Template-Seite mehrfach auf der Ausgabeseite zu platzieren, wobei deren Blöcke jedes Mal mit Daten gefüllt werden. So könnte das Template für eine Visitenkarte zum Beispiel vier Mal auf ein Blatt montiert werden. Die Blockfunktionen kümmern sich um Transformationen des Koordinatensystems und die korrekte Platzierung des Texts für alle Blöcke bei allen Aufrufen der Seite. Vorausgesetzt wird hier lediglich, dass der Client die Seite platziert und alle Blöcke auf der platzierten Seite verarbeitet. Die Seite kann dann auf der Ausgabeseite an anderer Stelle erneut platziert werden, wobei weitere Blockverarbeitung an der neuen Position stattfindet usw.

*Hinweis* Das Block-Plugin zeigt die Blockkoordinaten anders an, als sie intern in der PDF-Datei gespeichert werden. Während das Plugin mit der Acrobat-üblichen Notation arbeitet und den Koordinatenursprung in der linken oberen Ecke hat, beziehen sich die internen Koordinaten (die im Block gespeichert werden) auf die PDF-Konvention und haben den Ursprung in der linken unteren Ecke der Seite.

## 6.2.3 Was spricht gegen PDF-Formularfelder?

Der erfahrene Acrobat-Benutzer mag sich fragen, wozu für PDFlib ein neues Blockkonzept implementiert wird, statt die bewährten Formularfelder von PDF zu nutzen. Der entscheidende Unterschied besteht darin, dass PDF-Formularfelder in erster Linie dafür konzipiert wurden, vom Benutzer ausgefüllt zu werden, während PDFlib-Blöcke automatisch ausgefüllt werden. Für Anwendungen, die sowohl interaktives als auch automatisches Ausfüllen benötigen, bietet das Block-Plugin eine Funktion, die Formularfel-

der automatisch in Blöcke konvertiert (siehe Abschnitt 6.3.4, »Konvertieren von PDF-Formularfeldern in PDFlib-Blöcke«, Seite 172).

Wenngleich die beiden Konzepte in vielen Punkten übereinstimmen, bieten PDFlib-Blöcke gegenüber PDF-Formularfeldern doch einige Vorteile. Diese werden in Tabelle 6.1 aufgezeigt.


Tabelle 6.1 Vergleich zwischen PDF-Formularfeldern und PDFlib-Blöcken

<b>Funktion</b>	<b>PDF-Formularfelder</b>	<b>PDFlib-Blöcke</b>
Konzeption	für interaktiven Gebrauch	für automatisches Ausfüllen
typographische Funktionen (neben Schriftart und -größe)	–	Unterschneiden, Wort- und Zeichenabstand, Unterstreichen/Überstreichen/Durchstreichen
Schriftbehandlung	Schrifteinbettung	Schrifteinbettung, Einbettung von Untergruppen, Zeichensatz
Textformatierung	linksbündig, rechtsbündig, mittig	linksbündig, rechtsbündig, mittig, Blocksatz; verschiedene Formatierungsalgorithmen und Steuermöglichkeiten; Inline-Optionen können zur Steuerung der Textdarstellung verwendet werden
Wechsel der Schrift und anderer Textattribute innerhalb des Texts	–	ja
kombiniertes Ergebnis ist Bestandteil der PDF-Seitenbeschreibung	–	ja
Benutzer können kombinierte Feldinhalte editieren	ja	nein
Funktionalität erweiterbar	–	ja (selbstdefinierte Blockeigenschaften)
Verwendung von Bilddateien zum Füllen	–	BMP, CCITT, GIF, PNG, JPEG, JPEG2000, TIFF
Farbunterstützung	RGB	Graustufen, RGB, CMYK, Schmuckfarbe, Lab
PDF/X-Kompatibilität	–	ja (sowohl Template mit Blöcken als auch kombiniertes Ergebnis)
Eigenschaften von Text und Grafik können beim Füllen geändert werden	–	ja

## 6.3 Anlegen von PDFlib-Blöcken

### 6.3.1 Interaktive Blockerzeugung mit dem Block-Plugin

**Einsatz des PDFlib-Blockwerkzeugs.** Das Block-Plugin zur Erzeugung von PDFlib-Blöcken ist dem Acrobat-Formularwerkzeug recht ähnlich. Solange das Blockwerkzeug ausgewählt ist, sind alle Blöcke auf der Seite sichtbar. Wenn Sie ein anderes Acrobat-Werkzeug auswählen, werden die Blöcke ausgeblendet, sind aber nach wie vor vorhanden. Es gibt verschiedene Arten, das Blockwerkzeug auszuwählen:

- ▶ Sie klicken auf das Blocksymbol  in der Werkzeugleiste *Erweiterte Bearbeitung* (in Acrobat 5: Werkzeugleiste *Bearbeiten*).
- ▶ Sie wählen den Menübefehl *PDFlib Block, PDFlib Block-Werkzeug*.
- ▶ Sie drücken das Tastaturkürzel *P*. Stellen Sie sicher, dass unter *Bearbeiten, Grundeinstellungen..., [Allgemein...], Allgemein* das Kontrollkästchen »Zugriffstasten zum Zugriff auf Werkzeuge verwenden« aktiviert ist, was standardmäßig nicht der Fall ist (in Acrobat 5 nicht erforderlich).

**Anlegen und Ändern von Blöcken.** Nach der Auswahl des Blockwerkzeugs legen Sie einen Block an, indem Sie mit dem Fadenkreuz-Zeiger einfach an der gewünschten Position auf der Seite ein Rechteck geeigneter Größe aufziehen. Blöcke sind immer Rechtecke, deren Kanten parallel zu den Seitenrändern verlaufen. Bei der Erstellung eines neuen Blocks erscheint das Dialogfeld *PDFlib-Blockeigenschaften*, in dem Sie verschiedene Einstellungen vornehmen können (siehe Abschnitt 6.3.2, »Bearbeiten von Blockeigenschaften«, Seite 170). Der Blockname wird automatisch angelegt, kann aber nachträglich geändert werden. Er muss innerhalb einer Seite eindeutig sein. Im oberen Bereich legen Sie den Blocktyp auf *Text, Image* oder *PDF* fest. Die Registerkarten *Allgemein* und *Spezial* sind immer verfügbar. Die Registerkarten *Text, Image* und *PDF* werden nur bei Auswahl des entsprechenden Blocktyps eingeblendet. Die Registerkarte *Textflow* wird nur bei Blöcken vom Typ *Text* angezeigt, sofern die Eigenschaft *textflow* auf *true* gesetzt ist. Die Registerkarte *Tabs* (Tabulatorpositionen) ist nur verfügbar, wenn die Eigenschaft *horthabmethod* auf der Registerkarte *Textflow* auf *ruler* gesetzt ist.

*Hinweis* Benutzen Sie nach dem Hinzufügen neuer Blöcke oder Bearbeiten existierender Blöcke den Acrobat-Befehl »Speichern unter« (und nicht »Speichern«), um die Dateigröße zu minimieren.

*Hinweis* Wenn Sie mit dem Acrobat-Plugin *Enfocus PitStop* Dokumente bearbeiten, die PDFlib-Blöcke enthalten, erhalten Sie unter Umständen die Meldung »Dieses Dokument enthält anwendungsspezifische Informationen von PDFlib. Klicken Sie auf 'OK' zum Fortfahren oder 'Abbrechen' zum Beenden.« Sie brauchen dieser Meldung keine weitere Beachtung zu schenken und können problemlos OK drücken.

**Selektieren von Blöcken.** Blockoperationen wie Kopieren oder Verschieben beziehen sich auf alle selektierten Blöcke. Mit dem Blockwerkzeug lassen sich ein oder mehrere Blöcke wie folgt auswählen:

- ▶ Um einen einzelnen Block auszuwählen, klicken Sie ihn einfach mit der Maus an.
- ▶ Um die Selektion zu erweitern, klicken Sie bei gedrückter Umschalttaste auf einen weiteren Block.
- ▶ Um alle Blöcke auf der Seite auszuwählen, drücken Sie Strg-A (Windows) bzw. Cmd-A (Mac) oder *Bearbeiten, Alles auswählen*.

**Das Kontextmenü.** Sind einer oder mehrere Blöcke selektiert, können Sie das Kontextmenü zum schnellen Zugriff auf blockspezifische Funktionen nutzen (die sich auch im Menü *PDFlib Block* befinden). Zum Öffnen des Kontextmenüs klicken Sie mit der rechten Maustaste (Windows) oder klicken bei gedrückter Strg-Taste (Mac) auf den oder die selektierten Blöcke.

Um einen Block zu löschen, selektieren Sie ihn mit dem Blockwerkzeug und drücken die *Entf*-Taste. Alternativ dazu wählen Sie den Befehl *Ändern, Löschen* im Kontextmenü.

**Einstellen von Blockgröße und -position.** Mit dem Blockwerkzeug können Sie einen oder mehrere Blöcke an eine andere Position bewegen. Wenn Sie die Umschalttaste (Shift) während des Ziehens gedrückt halten, sind nur horizontale und vertikale Bewegungen möglich, was die exakte Ausrichtung von Blöcken erleichtert. Wenn Sie den Zeiger in die Nähe einer Blockecke bewegen, verwandelt er sich in einen Pfeil, mit dem Sie die Blockgröße ändern können. Um die Position oder Größe mehrerer Blöcke anzupassen, wählen Sie diese aus und wählen im Menü *PDFlib Block* oder im Kontextmenü die Befehle aus den Untermenüs *Ausrichten, Zentrieren, Verteilen* und *Skalieren*. Die Position eines oder mehrerer Blöcke lässt sich auch mit den Pfeiltasten ändern.

Alternativ dazu können Sie im Eigenschaftendialog numerische Blockkoordinaten eingeben. Der Ursprung des Koordinatensystems liegt in der linken oberen Ecke der Seite. Die Koordinaten werden in der Einheit angezeigt, die in Acrobat gerade eingestellt ist. Zum Ändern der Einheit gehen Sie wie folgt vor:

- ▶ In Acrobat 6/7 wählen Sie *Bearbeiten, Grundeinstellungen..., [Allgemein...], Einheiten und Hilfslinien* und selektieren Punkt, Millimeter, Zoll, Pica oder Zentimeter. Sie können auch *Anzeige, Navigationsregisterkarten, Info* wählen und eine Einheit aus dem Menü *Optionen* selektieren.
- ▶ In Acrobat 5 wählen Sie *Bearbeiten, Grundeinstellungen, Allgemein..., Anzeigen, Seiteneinheiten* und selektieren Punkt, Millimeter oder Zoll. Sie können auch *Fenster, Info* wählen und eine Einheit aus dem Menü *Info* selektieren.

Beachten Sie, dass die gewählte Einheit ausschließlich auf die Eigenschaft *Rect* und sonst keine numerische Eigenschaft angewandt wird.

**Anlegen von Blöcken durch Auswahl eines Rasterbildes oder einer Vektorgrafik.** Statt das Rechteck für einen Block manuell aufzuziehen, können Sie die Blockgröße auch anhand von vorhandenem Seiteninhalt definieren. Dazu müssen Sie zunächst sicherstellen, dass der Menüpunkt *PDFlib Block, Zur Blockdefinition Objekt anklicken* aktiv ist. Wenn Sie nun mit dem Blockwerkzeug auf ein Rasterbild auf der Seite klicken, wird ein Block in der Größe dieses Bildes erzeugt. Sie können ebenso auf andere grafische Objekte klicken. Das Blockwerkzeug versucht, die umgebende Grafik (zum Beispiel ein Logo) zu selektieren. Diese Objekt-Klick-Funktion ist als Hilfsmittel zur Definition von Blöcken gedacht. Sie können einen Block auch nachträglich ohne Einschränkungen verschieben oder in der Größe verändern. Der Block ist nicht an das Rasterbild oder das grafische Objekt gebunden, das als Hilfsmittel verwendet wurde.

Die Objekt-Klick-Funktion versucht zu erkennen, welche Vektorgrafiken oder Rasterbilder ein logisch zusammengehörendes Element auf der Seite bilden. Wenn Sie auf einen Seiteninhalt klicken, so wird dessen Boundingbox (das umschließende Rechteck) ermittelt und selektiert, sofern das Objekt nicht weiß oder sehr groß ist. Im nächsten Schritt werden weitere Objekte, die sich teilweise im ermittelten Rechteck befinden,

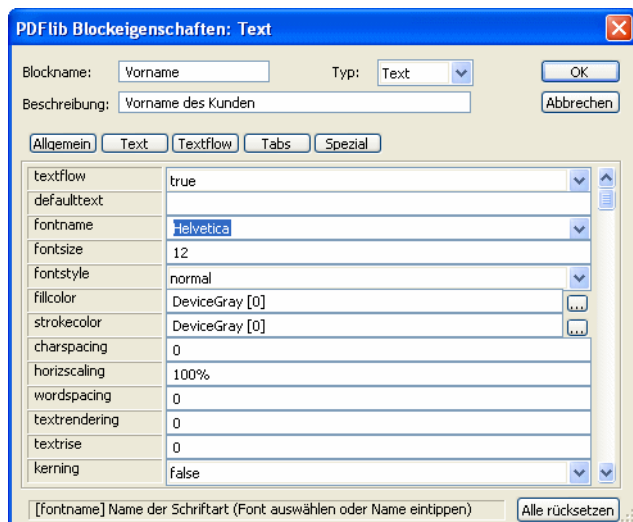


Abb. 6.1  
Einstellung der Blockeigenschaften; die Registerkarte Textflow erscheint nur bei `textflow=true`; die Registerkarte Tabs ist nur bei `hortabmethod=ruler` sichtbar

zum selektierten Bereich hinzugenommen und so weiter. Auf der Grundlage des endgültigen Bereichs wird das Blockrechteck generiert. Die Objekt-Klick-Funktion versucht letztendlich, vollständige Grafiken und nicht nur einzelne Linien zu selektieren.

Die Funktion ist aber nicht perfekt und selektiert unter Umständen (je nach Seiteninhalt) nicht immer das Gewünschte. Schließlich ist sie nur als Hilfsmittel zur schnellen Platzierung und Bemaßung von Rechtecken gedacht.

**Automatische Erkennung von Fonteigenschaften.** Das Block-Plugin ist in der Lage, den Font zu analysieren, der sich an der Stelle befindet, an der der Block positioniert wird. Darauf aufbauend kann es die folgenden entsprechenden Blockeigenschaften automatisch füllen:

fontname, fontsize, fillcolor, charspacing, horizscaling, wordspacing, textrendering, textrise

Da die automatische Erkennung von Fonteigenschaften zu unerwünschten Ergebnissen führen kann, wenn der Hintergrund ignoriert werden soll, kann diese Funktion mit *PDFlib Block, Automatische Font- und Farberkennung* aktiviert oder deaktiviert werden. Standardmäßig ist die Funktion deaktiviert.

**Sperren von Blöcken.** Blöcke können gegen versehentliches Bewegen, Verkleinern/Vergrößern oder Löschen gesperrt werden. Dazu wählen Sie das Blockwerkzeug, selektieren den gewünschten Block und wählen *Sperren* aus dem Kontextmenü. Ist ein Block gesperrt, lässt er sich weder bewegen, in der Größe ändern oder löschen noch können seine Eigenschaften angezeigt werden.

**Einsatz von Blöcken mit PDF/X.** Im Gegensatz zu PDF-Formularfeldern sind PDFlib-Blöcke kompatibel zu PDF/X. Sowohl das Eingabedokument, das Blöcke enthält, als auch die generierte PDF-Ausgabe lassen sich zu PDF/X kompatibel machen. Bei der Vorbereitung von Blockdateien für einen PDF/X-Workflow kann jedoch folgender Konflikt auftreten:

- ▶ PDF/X-1:2001, PDF/X-1a:2001 und PDF/X-3:2002 basieren auf Acrobat 4/PDF 1.3, so dass keine Acrobat-5-Dateien unterstützt werden;
- ▶ Das Block-Plugin setzt Acrobat 5 oder höher voraus.

Dieses Problem lässt sich beheben, indem die Dateien mit Acrobat in die erforderliche PDF-Version konvertiert werden. Weitere Informationen hierzu finden Sie im Abschnitt »PDF-Version eines Dokuments ändern«, Seite 190.

### 6.3.2 Bearbeiten von Blockeigenschaften

Beim Anlegen eines neuen Blocks, beim Doppelklick auf einen vorhandenen Block oder bei der Selektion von *Eigenschaften* im Kontextmenü des Blocks erscheint der Dialog *PDFlib Blockeigenschaften*, in dem Sie die Eigenschaften des selektierten Blocks einstellen können (siehe Abbildung 6.1). Wie in Abschnitt 6.4, »Standardeigenschaften zur automatischen Verarbeitung«, Seite 175, beschrieben, gibt es mehrere Arten von Eigenschaften:

- ▶ Blockname, Typ und Beschreibung sowie die Eigenschaften auf der Registerkarte *Allgemein* beziehen sich auf alle Blöcke.
- ▶ Die Eigenschaften auf den Registerkarten *Text*, *Image* und *PDF* beziehen sich nur auf den jeweiligen Blocktyp. Es wird immer nur die dem Blocktyp entsprechende Registerkarte eingeblendet, die anderen beiden Registerkarten sind inaktiv.
- ▶ Ist bei einem Textblock die Eigenschaft *textflow* auf *true* gesetzt, erscheint eine weitere Registerkarte namens *Textflow*, die textfluss-spezifische Einstellungen enthält.
- ▶ Ist bei einem Textblock die Eigenschaft *textflow* auf *true* und die Eigenschaft *hortabmethod* auf der Registerkarte *Textflow* auf *ruler* gesetzt, erscheint noch zusätzlich die Registerkarte *Tabs*, auf der sich die Tabulatoren einstellen lassen.
- ▶ Die Eigenschaften auf der Registerkarte *Spezial* können vom Benutzer definiert werden und beziehen sich auf einen beliebigen Blocktyp.

Um den Wert einer Eigenschaft zu ändern, geben Sie die gewünschte Zahl oder den gewünschten String in das Eingabefeld der Eigenschaft ein (z.B. bei *linewidth*), selektieren den Wert aus einer Dropdown-Liste (z.B. bei *fontname*, *fitmethod*) oder selektieren mit dem »...«-Button rechts einen Farbwert oder Dateinamen (z.B. bei *backgroundcolor*). Bei der Eigenschaft *fontname* können Sie die Schrift aus einer Liste mit allen im System installierten Schriften auswählen oder einen Schriftnamen eingeben. Unabhängig von der Auswahlmethode muss der gewählte Font auf dem System vorhanden sein, auf dem die Blöcke mit dem PDFlib Personalization Server gefüllt werden.

Nachdem Sie die gewünschten Einstellungen vorgenommen haben, schließen Sie den *Eigenschaften*-Dialog durch Klicken auf OK. Die soeben definierten Eigenschaften werden als Bestandteil der Blockdefinition in der PDF-Datei gespeichert.

**Übereinander liegende Blöcke.** Überlappende Blöcke lassen sich oft nur schwer selektieren, da beim Klicken in einen Bereich nur der oberste Block selektiert wird. In solchen Fällen kann der Befehl *Block auswählen* im Kontextmenü genutzt werden, um einen der Blöcke anhand seines Namens zu selektieren. Die nächste Aktion, die im Bereich des selektierten Blocks durchgeführt wird (z.B. ein Doppelklick), bezieht sich dann nur auf diesen Block. Auf diese Weise lassen sich Blöcke problemlos bearbeiten, selbst wenn sie teilweise oder vollständig von anderen Blöcken verdeckt werden.

**Verwendung und Wiederherstellung von Standardwerten.** Um sich Tipp- und Klickaufwand zu ersparen, merkt sich das Blockwerkzeug die Werte, die im *Eigenschaften*-Dialog für den zuletzt definierten Block eingegeben wurden. Diese werden beim Anlegen eines neuen Blocks wiederverwendet. Natürlich können Sie diese Werte jederzeit umdefinieren.

Mit dem Button *Alle rücksetzen* im Fenster *PDFlib Blockeigenschaften* werden die meisten Eigenschaften auf ihre Standardwerte zurückgesetzt. Folgende Eigenschaften bleiben unverändert:

- ▶ Die Eigenschaften *Blockname*, *Typ*, *Beschreibung* und *Rect*.
- ▶ Alle selbstdefinierten Eigenschaften auf der Registerkarte *Spezial*.

**Gemeinsame Eigenschaften.** Mit dem Blockwerkzeug können Sie bei gedrückter Umschalttaste mehrere Blöcke auf der Seite selektieren. Wenn Sie dann auf einen der Blöcke doppelklicken oder die Eingabetaste drücken, öffnet sich der *Eigenschaften*-Dialog, der sich jetzt auf alle selektierten Blöcke bezieht. Es können jedoch nur solche Eigenschaften editiert werden, die auf alle selektierten Blöcke anwendbar sind. Abschnitt 6.4, »Standardeigenschaften zur automatischen Verarbeitung«, Seite 175, beschreibt, welche Eigenschaften mehreren Blöcken gemeinsam sein können. Selbstdefinierte Eigenschaften können nicht gemeinsam genutzt werden.

### 6.3.3 Kopieren von Blöcken zwischen Seiten und Dokumenten

Das Block-Plugin bietet mehrere Methoden zum Verschieben oder Kopieren von Blöcken auf der aktuellen Seite, innerhalb des aktuellen Dokuments oder zwischen Dokumenten:

- ▶ Blöcke können durch Ziehen mit der Maus oder durch Einfügen auf einer anderen Seite oder in ein anderes offenes Dokument verschoben oder kopiert werden.
- ▶ Blöcke können auf eine oder mehrere Seiten desselben Dokuments dupliziert werden.
- ▶ Blöcke können in eine neue Datei (mit leeren Seiten) oder in ein vorhandenes Dokument (auf die vorhandenen Seiten) exportiert werden.
- ▶ Blöcke können aus anderen Dokumenten importiert werden.

Um den Seiteninhalt unter Beibehaltung der Blockdefinitionen zu aktualisieren, können Sie die zugrunde liegenden Seiten ohne Veränderung der Blöcke ersetzen. Dazu verwenden Sie *Dokument, Seiten ersetzen...* (Acrobat 5 und 7) oder *Dokument, Seiten, Ersetzen* (Acrobat 6).

**Verschieben und Kopieren von Blöcken.** Sie können Blöcke verschieben oder kopieren, indem Sie einen oder mehrere Blöcke selektieren und bei gedrückter Strg-Taste (Windows) bzw. Alt-Taste (Mac) an eine neue Position ziehen. So lange die Taste gedrückt ist, nimmt der Mauszeiger eine andere Form an. Ein kopierter Block hat die gleichen Eigenschaften wie der ursprüngliche Block mit Ausnahme des Namens, der automatisch geändert wird.

Ebenso können Sie Blöcke mit Copy&Paste an eine andere Position auf der selben Seite, auf einer anderen Seite des selben Dokuments oder eines anderen in Acrobat geöffneten Dokuments kopieren:

- ▶ Wählen Sie das Blockwerkzeug und selektieren Sie die zu kopierenden Blöcke.
- ▶ Kopieren Sie die selektierten Blöcke mit Strg-C (Windows) bzw. Cmd-C (Mac) oder *Bearbeiten, Kopieren* in die Zwischenablage.

- ▶ Fügen Sie mit Strg-V (Windows) bzw. Cmd-V (Mac) oder *Bearbeiten, Einfügen* die Blöcke aus der Zwischenablage ein.

**Duplizieren von Blöcken auf andere Seiten.** Sie können einen oder mehrere Blöcke gleichzeitig auf eine beliebige Anzahl von Seiten im Dokument duplizieren:

- ▶ Wählen Sie das Blockwerkzeug und selektieren Sie die zu duplizierenden Blöcke.
- ▶ Wählen Sie *Import und Export, Duplizieren...* im Menü *PDFlib Block* oder im Kontextmenü.
- ▶ Wählen Sie aus, ob nur die selektierten oder alle Blöcke auf der Seite und auf welchen Seitenbereich die Blöcke dupliziert werden sollen.

**Import und Export von Blöcken.** Mit den Funktionen zum Import und Export von Blöcken können alle Blockdefinitionen auf der Seite oder in einem Dokument über mehrere PDF-Dateien verteilt werden. Dies ist zum Beispiel bei der Aktualisierung des Seiteninhalts sinnvoll, wenn vorhandene Blockdefinitionen erhalten bleiben sollen. Um die Blockdefinitionen in eine eigene Datei zu exportieren, gehen Sie wie folgt vor:

- ▶ Wählen Sie das Blockwerkzeug und selektieren Sie die zu exportierenden Blöcke.
- ▶ Wählen Sie den Befehl *Import und Export, Export...* im Menü *PDFlib Block* oder im Kontextmenü.
- ▶ Geben Sie den Seitenbereich sowie einen Namen für die Datei ein, die die Blockdefinitionen enthalten soll.

Mit dem Befehl *Import und Export, Import...* im Menü *PDFlib Block* oder im Kontextmenü importieren Sie Blockdefinitionen. Dabei können Sie auswählen, ob die importierten Blöcke auf alle Seiten oder nur einen Seitenbereich im Dokument platziert werden. Bei mehreren selektierten Seiten werden die Blockdefinitionen unverändert auf diese kopiert. Enthält der Seitenbereich des Zieldokuments mehr Seiten als die Datei mit den zu importierenden Blockdefinitionen, können Sie die Checkbox *Vorlage wiederholen* selektieren. Die Blöcke in der zu importierenden Datei werden dann so lange wiederholt auf das Zieldokument übertragen, bis das Dokumentende erreicht ist.

**Kopieren von Blöcken in ein anderes Dokument beim Export.** Beim Exportieren von Blöcken können Sie diese unmittelbar auf die Seiten eines anderen Dokuments übertragen. Dazu wählen Sie ein bereits existierendes Dokument als Exportdatei. Wenn Sie die Checkbox *Vorhandene Blöcke löschen* selektieren, werden alle im Zieldokument bereits vorhandenen Blöcke gelöscht, bevor die neuen Blöcke in das Dokument kopiert werden.

### 6.3.4 Konvertieren von PDF-Formularfeldern in PDFlib-Blöcke

Statt PDFlib-Blöcke manuell zu erstellen, können Sie PDF-Formularfelder auch automatisch in PDFlib-Blöcke konvertieren. Dies ist insbesondere dann von Vorteil, wenn Sie bereits über komplexe PDF-Formulare mit zahlreichen Feldern verfügen, die in Zukunft automatisch mit dem PDFlib Personalization Server gefüllt werden sollen oder eine große Anzahl vorhandener PDF-Formulare umwandeln müssen, damit sie automatisch ausfüllbar sind. Um alle Formularfelder auf einer Seite in PDFlib-Blöcke zu konvertieren, wählen Sie *PDFlib Block, Formularfelder konvertieren, Aktuelle Seite*. Um alle Formularfelder in einem Dokument zu konvertieren, verwenden Sie stattdessen *Alle Seiten*. Um nur die selektierten Formularfelder (zur Selektion wählen Sie das Formular-Werkzeug oder das Objektauswahl-Werkzeug von Acrobat) zu konvertieren, verwenden Sie *Ausgewählte Formularfelder*.

**Konvertierung einzelner Eigenschaften.** Bei der automatischen Konvertierung von Formularfeldern werden Formularfelder der im Dialogfeld *PDFlib Block, Formularfelder konvertieren, Konvertierungseinstellungen...* in Blöcke vom Typ *Text* umgewandelt. Standardmäßig werden alle Formularfeldtypen konvertiert. Die Eigenschaften der Formularfelder werden gemäß Tabelle 6.2 in entsprechende Blockeigenschaften umgewandelt.

Tabelle 6.2 Konvertierung von PDF-Formularfeldern in PDFlib-Blöcke

<b>PDF-Formularfeldeigenschaft...</b>	<b>...wird konvertiert in PDFlib-Blockeigenschaft</b>
<b>Alle Felder</b>	
Position	Allgemein, Rect
Name	Allgemein, Name
Tooltip	Allgemein, Description
Darstellung, Text, Schrift	Text, fontname
Darstellung, Text, Schriftgröße	Text, fontsize; die Schriftgröße »auto« wird in eine feste Größe von 2/3 der Blockhöhe konvertiert, außerdem wird für die Eigenschaft fitmethod der Wert »auto« eingetragen. Bei mehrzeiligen Feldern/Blöcken ergibt diese Kombination eine passende Schriftgröße, die kleiner als der Anfangswert von 2/3 der Blockhöhe sein kann.
Darstellung, Text, Farbe	Text, textcolor und Text, fillcolor
Darstellung, Umrandung und Farbe, Umrandungsfarbe	Allgemein, bordercolor
Darstellung, Umrandung und Farbe, Füllfarbe	Allgemein, backgroundcolor
Darstellung, Umrandung und Farbe, Linienstärke	Allgemein, linewidth: Thin=1, Medium=2, Thick=3
Allgemein, Allgemeine Eigenschaften, Formularfeld	Allgemein, Status: Sichtbar=active; Unsichtbar=ignore; Sichtbar, aber Drucken nicht möglich=ignore; Unsichtbar, aber Drucken ist möglich=active
Allgemein, Allgemeine Eigenschaften, Ausrichtung	Allgemein, orientate: 0=north, 90=west, 180=south, 270=east
<b>Textfelder</b>	
Optionen, Standardwert	Text, defaulttext
Optionen, Ausrichtung	Allgemein, position: Links={0 50}, Zentriert={50 50}, Rechts={100, 50}
Optionen, Mehrere Zeilen	Text, textflow: aktiviert=true, nicht aktiviert=false
<b>Optionsfelder und Kontrollkästchen</b>	
Falls »Schaltfläche ist standardmäßig aktiviert« angeklickt ist: Optionen, Schaltflächenstil bzw. Optionen, Kontrollkästchenstil	Text, defaulttext: Häkchen=4, Kreis=I, Kreuz=8, Karo=u, Quadrat=n, Stern=H (diese Zeichen stellen die jeweiligen Symbole im Font ZapfDingbats dar)
<b>Kombinationsfelder und Listenfelder</b>	
Optionen, ausgewähltes Standardelement	Text, defaulttext
<b>Schaltflächen</b>	
Optionen, Symbol und Beschriftung, Beschriftung	Text, defaulttext

**Formularfelder mit gleichen Namen.** Gleichnamige Formularfelder auf der Seite sind erlaubt, Blocknamen dagegen müssen auf einer Seite eindeutig sein. Bei der Formular-konvertierung werden deshalb Zahlen an die generierten Blocknamen angehängt, um diese eindeutig zu machen.

Beachten Sie, dass aufgrund eines Fehlers in Acrobat die Attribute von Formularfeldern gleichen Namens nicht korrekt wiedergegeben werden. Haben mehrere Felder denselben Namen, aber unterschiedliche Attribute, werden diese Unterschiede nicht korrekt in die generierten Blöcke übernommen. Bei der Konvertierung erscheint in solchen Fällen eine Warnung mit den Namen der betroffenen Formularfelder. Sie sollten die Eigenschaften in den generierten Blöcken dann genau überprüfen.

**Binden von Blöcken an zugehörige Formularfelder.** Um PDF-Formularfelder und die daraus generierten PDFlib-Blöcke aufeinander abgestimmt zu halten, können die generierten Blöcke an die entsprechenden Formularfelder gebunden werden. Das Blockwerkzeug erhält dann die Beziehung zwischen Formularfeldern und Blöcken aufrecht. Wird der Konvertierungsprozess erneut durchgeführt, so wird ein gebundener Block gemäß der Eigenschaften des zugehörigen PDF-Formularfeldes aktualisiert. Gebundene Blöcke verhindern, dass dieselbe Arbeit doppelt erledigt werden muss: Bei der Aktualisierung eines Formulars zur interaktiven Verwendung kann der entsprechende Block automatisch mit aktualisiert werden.

Wenn Sie die Formularfelder nach der Konvertierung nicht mehr benötigen, wählen Sie im Dialogfeld *PDFlib Block, Formularfelder konvertieren, Konvertierungseinstellungen...* die Option *Konvertierte Formularfelder löschen*. Bei dieser Option werden die Formularfelder nach der Konvertierung gelöscht. Alle Aktionen (zum Beispiel JavaScript), die den betroffenen Feldern zugeordnet sind, werden ebenfalls aus dem Dokument entfernt.

**Stapelkonvertierung.** Wenn Sie die Formularfelder vieler PDF-Dokumente in PDFlib-Blöcke konvertieren möchten, können Sie die Stapelkonvertierung nutzen, die automatisch eine beliebige Anzahl von Dokumenten verarbeitet. Der Dialog zur Stapelverarbeitung kann über *PDFlib Block, Formularfelder konvertieren, Stapelkonvertierung...* geöffnet werden:

- ▶ Es können einzelne Dateien oder der vollständige Inhalt eines Verzeichnisses zur Verarbeitung ausgewählt werden.
- ▶ Die Ausgabedateien können im Verzeichnis der Eingabedateien oder in einem anderen Verzeichnis abgelegt werden. Sie können mit einem Präfix versehen werden, um sie von den Eingabedateien zu unterscheiden.
- ▶ Bei der Verarbeitung sehr vieler Dokumente sollte eine Log-Datei angegeben werden. In dieser werden alle verarbeiteten Dateien aufgelistet und zu jeder Datei Einzelheiten zur Konvertierung einschließlich eventueller Fehlermeldungen protokolliert.

Während der Konvertierung sind die PDF-Dokumente in Acrobat sichtbar, es können aber keinerlei Acrobat-Menüfunktionen oder Werkzeuge verwendet werden.

## 6.4 Standardeigenschaften zur automatischen Verarbeitung

PDFlib bietet einige allgemeine Eigenschaften (»Properties«), die jedem Blocktyp zugeordnet werden können. Daneben gibt es für den jeweiligen Blocktyp *Text*, *Image* und *PDF* spezifische Eigenschaften. Manche Eigenschaften können gemeinsam genutzt werden, das heißt, dass sie mit dem Block-Plugin mehreren Blöcken auf einmal zugeordnet werden können.

Eigenschaften unterstützen dieselben Datentypen wie Optionslisten (siehe Abschnitt 3.1.4, »Optionslisten«, Seite 51) mit Ausnahme von Handles und Aktionslisten.

Viele Blockeigenschaften heißen wie die Optionen für *PDF\_fit\_image()* und andere Funktionen (zum Beispiel *fitmethod*) oder wie PDFlib-Parameter (zum Beispiel *charspacing*). In solchen Fällen verhält sich die Eigenschaft genau so wie die gleichnamige Option bzw. der Parameter.

**Verarbeitung von Eigenschaften in PDFlib.** Die PDFlib-Blockfunktionen *PDF\_fill\_block()* verarbeiten Blockeigenschaften in der folgenden Reihenfolge:

- ▶ Ist die Eigenschaft *backgroundcolor* vorhanden, wird das Blockrechteck mit der festgelegten Farbe gefüllt.
- ▶ Alle anderen Eigenschaften mit Ausnahme von *bordercolor* und *linewidth* werden verarbeitet.
- ▶ Ist die Eigenschaft *bordercolor* vorhanden, wird das Blockrechteck mit der festgelegten Farbe und Linienstärke gezeichnet.

Es wird nichts beschnitten; um sicherzustellen, dass der Blockinhalt nicht über das Blockrechteck hinausreicht, wählen Sie *fitmethod nofit*.

Wird in einer Blockeigenschaft eine Schmuckfarbe verwendet, muss der definierte Schmuckfarbname entweder PDFlib-intern bekannt sein (siehe Abschnitt 3.3.3, »Schmuckfarben«, Seite 68) oder er muss im PDFlib-Clientprogramm bereits früher mit *PDF\_makespotcolor()* festgelegt worden sein. Andernfalls scheitern die Blockfunktionen.

### 6.4.1 Allgemeine Eigenschaften

Allgemeine Eigenschaften beziehen sich auf alle Arten von Blöcken (*Text*, *Image*, *PDF*). Sie sind zur Blockverwaltung erforderlich, beschreiben das Aussehen des Blockrechtecks und legen fest, wie der Inhalt im Block platziert wird. Alle erforderlichen Einträge werden vom Block-Plugin automatisch generiert. Tabelle 6.3 gibt eine Übersicht über die allgemeinen Eigenschaften.

Tabelle 6.3 Allgemeine Blockeigenschaften

Schlüsselwort	Typ	Mögliche Werte und Erklärung
<b>Blockverwaltung</b>		
Name	String	(Erforderlich) Name des Blocks. Blocknamen müssen auf der Seite, aber nicht innerhalb des Dokuments eindeutig sein. Die drei Zeichen [ ] / sind in Blocknamen nicht erlaubt. Blocknamen dürfen maximal 127 Zeichen lang sein.
Description	String	Vom Benutzer lesbare Beschreibung der Funktion des Blocks, die in PDFDocEncoding oder Unicode kodiert ist (und im letzteren Fall mit einem BOM beginnt). Diese Eigenschaft dient nur zur Information des Benutzers und wird bei der Blockverarbeitung ignoriert.

Tabelle 6.3 Allgemeine Blockeigenschaften

Schlüsselwort	Typ	Mögliche Werte und Erklärung
Locked	Boolean	(Gemeinsam nutzbar) Ist diese Eigenschaft gleich true, lassen sich der Block und seine Eigenschaften nicht mit dem Block-Plugin bearbeiten. Diese Eigenschaft wird bei der Blockverarbeitung ignoriert. Standardwert: false
Rect	Float-Liste	(Erforderlich) Blockkoordinaten. Der Ursprung des Koordinatensystems liegt in der linken unteren Ecke der Seite. Das Block-Plugin zeigt die Koordinaten aber in der Notation von Acrobat an, das heißt, mit dem Ursprung in der linken oberen Ecke der Seite. Die Koordinaten werden in der Einheit angezeigt, die in Acrobat gerade ausgewählt ist. In der PDF-Datei werden sie immer in Punkt gespeichert.
Status	Schlüsselwort	Schlüsselwort, das beschreibt, wie der Block verarbeitet wird. Standardwert: active active Der Block wird entsprechend seiner Eigenschaften komplett verarbeitet. ignore Der Block wird ignoriert. static Es wird kein variabler Inhalt platziert, sondern der Standardblockinhalt in Form von Text, Image oder PDF-Inhalt verwendet, sofern vorhanden.
Subtype	Schlüsselwort	(Erforderlich) Je nach Blocktyp entweder Text, Image oder PDF
Type	Schlüsselwort	(Erforderlich) Immer Block
<b>Blockdarstellung</b>		
background-color	Color	(Gemeinsam nutzbar) Ist diese Eigenschaft vorhanden und enthält sie ein von None verschiedenes Schlüsselwort zur Festlegung des Farbraums, wird ein Rechteck gezeichnet und mit der angegebenen Farbe gefüllt. Dies kann nützlich sein, um vorhandenen Seiteninhalt zu verdecken. Standardwert: None.
bordercolor	Color	(Gemeinsam nutzbar) Ist diese Eigenschaft vorhanden und enthält sie ein von None verschiedenes Schlüsselwort zur Festlegung des Farbraums, wird ein Rechteck mit einem Rand in der angegebenen Farbe gezeichnet. Standardwert: None
linewidth	Float	(Gemeinsam nutzbar, muss größer 0 sein) Breite der Linie, mit der das Blockrechteck gezeichnet wird; wird nur verwendet, wenn bordercolor gesetzt ist. Standardwert: 1
<b>Platzierung des Inhalts</b>		
fitmethod	Schlüsselwort	(Gemeinsam nutzbar) Strategie für den Fall, dass der übergebene Inhalt nicht in die Box passt. Mögliche Werte sind auto, nofit, clip, meet <sup>1</sup> , slice <sup>1</sup> und entire <sup>1</sup> . Für einfache Blöcke vom Typ text, image oder PDF wird diese Eigenschaft gemäß Tabelle 8.18 und Tabelle 8.39 interpretiert. Standardwert: auto. Für Textflussblöcke, die zu schmal für den Text sind, lautet die Interpretation wie folgt: auto fontsize und leading werden so lange reduziert, bis der Text passt. nofit Der Text läuft aus dem unteren Blockrand hinaus. clip Der Text wird am Blockrand abgeschnitten.
orientate	Schlüsselwort	(Gemeinsam nutzbar) Legt fest, in welcher Ausrichtung der Inhalt platziert wird (siehe Tabelle 8.39). Mögliche Werte sind north, east, south, west. Standardwert: north
position <sup>1</sup>	Float-Liste	(Gemeinsam nutzbar) Einer oder zwei Werte, die die Position des Referenzpunkts innerhalb des Inhalts festlegen (siehe Tabelle 8.18 für text und Tabelle 8.39 für image/PDF). Standardwert: 0
rotate	Float	(Gemeinsam nutzbar) Drehwinkel in Grad, um den der Block gegen den Uhrzeigersinn gedreht wird, bevor die Verarbeitung beginnt. Das Rotationszentrum ist der Referenzpunkt. Standardwert: 0

1. Dieses Schlüsselwort bzw. diese Eigenschaft wird für Textflussblöcke (Blöcke vom Typ Text mit textflow=true) nicht unterstützt.

## 6.4.2 Textspezifische Eigenschaften

Für Blöcke vom Typ `Text` können neben den allgemeinen Eigenschaften einige textspezifische Eigenschaften definiert werden. Alle textspezifischen Eigenschaften lassen sich gemeinsam nutzen.

**Generelle Eigenschaften für Textblöcke.** Textblöcke können abhängig von der Eigenschaft `textflow` aus einer oder mehreren Zeilen bestehen. Tabelle 6.4 zeigt alle Eigenschaften, die sich generell auf beide Textblockarten anwenden lassen.

Tabelle 6.4 Textspezifische Blockeigenschaften

Schlüsselwort	Typ	Mögliche Werte und Erklärung
<code>charspacing</code>	Float oder Prozentwert	Zeichenabstand (siehe Tabelle 8.17). Prozentangaben beziehen sich auf <code>fontsize</code> . Standardwert: 0
<code>defaulttext</code>	String	Text, der verwendet wird, wenn vom Client kein Ersatztext übergeben wird <sup>1</sup>
<code>fillcolor</code>	Color	Füllfarbe des Texts. Standardwert: <code>gray 0</code> (=schwarz)
<code>fontname</code> <sup>2</sup>	String	Name der Schrift, so wie bei <code>PDF_load_font()</code> erforderlich. Das Block-Plugin zeigt alle installierten Systemschriften in einer Liste an. Beachten Sie jedoch, dass diese Fontnamen nicht unbedingt zwischen Mac, Windows und Unix portierbar sind. Der Encoding für den Text muss beim Füllen der Blöcke als Option für <code>PDF_fill_textblock()</code> angegeben werden, es sei denn, die Option <code>font</code> wurde benutzt.
<code>fontsize</code> <sup>2</sup>	Float	Schriftgröße in Punkt
<code>fontstyle</code>	Schlüsselwort	Schriftstil; mögliche Schlüsselwörter sind <code>normal</code> , <code>bold</code> , <code>italic</code> oder <code>bolditalic</code> (siehe Tabelle 8.15)
<code>horizscaling</code>	Float oder Prozentwert	Horizontale Skalierung von Text (siehe Tabelle 8.17). Standardwert: 100%
<code>italicangle</code>	Float	Neigungswinkel von kursivem Text in Grad (siehe Tabelle 8.17). Standardwert: 0
<code>kerning</code>	Boolean	Unterschneidung (siehe Tabelle 8.17). Standardwert: <code>false</code>
<code>margin</code>	Float-Liste	1 oder 2 Float-Werte für eine zusätzliche horizontale und vertikale Ausdehnung der Textbox (siehe Tabelle 8.18). Standardwert: 0
<code>monospace</code>	Integer 1...2048	Alle Glyphen des Fonts werden in äquidistanter Breite geschrieben (siehe Tabelle 8.15). Standardwert: nicht vorhanden (es werden die Metrikdaten der Schrift verwendet)
<code>overline</code>	Boolean	Modus für Überstreichen (siehe Tabelle 8.17). Standardwert: <code>false</code>
<code>strikeout</code>	Boolean	Modus für Durchstreichen (siehe Tabelle 8.17). Standardwert: <code>false</code>
<code>strokecolor</code>	Color	Farbe, in der Text gezeichnet wird. Standardwert: <code>gray 0</code> (= schwarz)
<code>textflow</code>	Boolean	Steuert die ein- oder mehrzeilige Verarbeitung (Standardwert: <code>false</code> ): <code>false</code> Text muss einzeilig sein und wird mit <code>PDF_fit_text()</code> verarbeitet. <code>true</code> Text kann mehrzeilig sein und wird mit <code>PDF_fit_textflow()</code> verarbeitet. Die allgemeine Eigenschaft <code>position</code> wird ignoriert. Neben den Standardeigenschaften für Text können alle Textflow-spezifischen Eigenschaften festgelegt werden (siehe Tabelle 6.5).
<code>textrendering</code>	Integer	Darstellungsmodus für Text (siehe Tabelle 8.17). Standardwert: 0
<code>textrise</code>	Float oder Prozentwert	Modus für den vertikalen Textversatz (siehe Tabelle 8.17). Prozentwerte beziehen sich auf <code>fontsize</code> . Standardwert: 0
<code>underline</code>	Boolean	Modus für Unterstreichen (siehe Tabelle 8.17). Standardwert: <code>false</code>
<code>underline-position</code>	Float, Prozentwert o. Schlüsselwort	Position der Linie für unterstrichenen Text, relativ zur Grundlinie (siehe Tabelle 8.19). Prozentwerte beziehen sich auf <code>fontsize</code> . Standardwert: <code>auto</code>

Tabelle 6.4 Textspezifische Blockeigenschaften

Schlüsselwort	Typ	Mögliche Werte und Erklärung
<code>underline-width</code>	Float, Prozentwert o. Schlüsselwort	Stärke der Linie für unterstrichenen Text (siehe Tabelle 8.19). Prozentwerte beziehen sich auf <code>fontsize</code> . Standardwert: <code>auto</code>
<code>wordspacing</code>	Float oder Prozentwert	Wortabstand (siehe Tabelle 8.17). Prozentwerte beziehen sich auf <code>fontsize</code> . Standardwert: <code>0</code>

1. Text wird im Zeichensatz `winansi` oder `Unicode` interpretiert.

2. Diese Eigenschaft muss für einen Textblock definiert sein; sie wird vom `PDFlib` Block-Plugin automatisch gesetzt.

**Eigenschaften für Textflussblöcke.** Textfluss-spezifische Eigenschaften beziehen sich auf Blöcke vom Typ `Text`, bei denen die Eigenschaft `textflow` den Wert `true` hat. Anhand der text-spezifischen Eigenschaften wird die anfängliche Optionsliste zur Textflow-Verarbeitung zusammengestellt (entsprechend dem Parameter `optlist` für `PDF_create_textflow()`). Mit dem Block-Plugin können keine Inline-Optionslisten festgelegt werden. Diese lassen sich jedoch auf dem Server als Bestandteil des Textinhalts übergeben, wenn der Block mit `PDF_fill_textblock()` gefüllt wird. Alle textfluss-spezifischen Eigenschaften lassen sich gemeinsam nutzen. Tabelle 6.5 gibt eine Übersicht.

Tabelle 6.5 Textfluss-spezifische Blockeigenschaften

Schlüsselwort	Typ	Mögliche Werte und Erklärung
<b>Optionen zur Textinterpretation</b>		
<code>tabalignchar</code>	Integer	Unicode-Wert des Zeichens, an dem dezimale Tabulatoren ausgerichtet werden. Standardwert: das Zeichen <code>'</code> ( <code>U+002E</code> )
<b>Optionen zur Steuerung der Textformatierung</b>		
<code>alignment</code>	Schlüsselwort	Legt die Formatierung für die Zeilen eines Absatzes fest. Standardwert: <code>left</code> <code>left</code> linksbündig, beginnend bei <code>leftindent</code> <code>center</code> mittig zwischen <code>leftindent</code> und <code>rightindent</code> <code>right</code> rechtsbündig, bei <code>rightindent</code> endend <code>justify</code> links- und rechtsbündig (Blocksatz)
<code>firstlinedist</code>	Float, Prozentwert o. Schlüsselwort	Abstand zwischen dem oberen Rand der Fitbox und der Grundlinie der ersten Textzeile. Angegeben wird er in Benutzerkoordinaten, als Prozentsatz der Schriftgröße (der Größe der ersten in der Zeile auftretenden Schrift, wenn <code>fixedleading=true</code> , oder der maximal auftretenden Schriftgröße andernfalls) oder als Schlüsselwort. Standardwert: <code>leading</code> . <code>leading</code> Für die erste Zeile ermittelter Zeilenabstand; diakritische Zeichen wie Å berühren dabei den oberen Rand der Fitbox. <code>ascender</code> Für die erste Zeile ermittelte Oberlänge; Zeichen mit großer Oberlänge wie <code>d</code> oder <code>h</code> berühren dabei den oberen Rand der Fitbox. <code>capheight</code> Für die erste Zeile ermittelte Versalhöhe; hohe Großbuchstaben wie <code>H</code> berühren dabei den oberen Rand der Fitbox. Ist <code>fixedleading=false</code> , wird der größte Wert verwendet, der für Zeilenabstand, Oberlänge und Versalhöhe in der ersten Zeile ermittelt wurde.
<code>fixedleading</code>	Boolean	Ist <code>fixedleading</code> gleich <code>true</code> , wird der beim ersten Zeichen geltende Zeilenabstand verwendet. Andernfalls wird der größte Zeilenabstand verwendet. Standardwert: <code>false</code> .
<code>hortabsize</code>	Float oder Prozentwert	Legt die Breite eines horizontalen Tabulators fest <sup>1</sup> . Die Interpretation wird von der Option <code>hortabmethod</code> beeinflusst. Standardwert: <code>7,5%</code> .

Tabelle 6.5 Textfluss-spezifische Blockeigenschaften

<b>Schlüsselwort</b>	<b>Typ</b>	<b>Mögliche Werte und Erklärung</b>
<i>hortab-method</i>	Schlüsselwort	Legt die Interpretation von horizontalen Tabulatoren im Text fest. Liegt die festgelegte Position links der aktuellen Textposition, so wird der Tabulator ignoriert. Standardwert: <i>relative</i> . <i>relative</i> Die Position wird um den in <i>hortabsize</i> festgelegten Betrag vorgerückt. <i>typewriter</i> Die Position wird auf das nächste Vielfache von <i>hortabsize</i> vorgerückt. <i>ruler</i> Die Position wird auf den <i>n</i> -ten in der Option <i>ruler</i> verfügbaren Tabulatorwert gesetzt, wobei <i>n</i> die Anzahl der bislang in der Textzeile vorgekommenen Tabs bezeichnet. Ist <i>n</i> größer als die Anzahl der in <i>ruler</i> verfügbaren Tabulatorpositionen, kommt die Methode <i>relative</i> zum Einsatz.
<i>lastalignment</i>	Schlüsselwort	Bestimmt die Formatierung der letzten Zeile eines Absatzes. Neben allen Schlüsselwörtern der Option <i>alignment</i> gibt es folgende Schlüsselwörter. Standardwert: <i>auto</i> . <i>auto</i> Es wird der Wert der Option <i>alignment</i> verwendet. Nur bei <i>justify</i> wird <i>left</i> verwendet.
<i>lastlinedist</i>	Float, Prozentwert o. Schlüsselwort	(Wird ignoriert bei <i>fitmethod=nofit</i> ) Der kleinste Abstand zwischen der Grundlinie der letzten Textzeile und dem unteren Rand der Fitbox. Angegeben wird er in Benutzerkoordinaten, als Prozentsatz der Schriftgröße (der ersten in der Zeile auftretenden Schriftgröße, wenn <i>fixedleading=true</i> oder der maximal in der Zeile auftretenden Schriftgröße andernfalls) oder als Schlüsselwort. Standardwert: <i>o</i> , d.h. der untere Rand der Fitbox wird als Grundlinie verwendet und die üblichen Unterlängen reichen aus der Fitbox hinaus. <i>descender</i> Für die letzte Zeile ermittelte Unterlänge; Zeichen mit Unterlängen wie <i>g</i> oder <i>j</i> berühren dabei den unteren Rand der Fitbox. Ist <i>fixedleading=false</i> wird der größte Wert verwendet, der in der letzten Zeile für die Unterlänge ermittelt wurde.
<i>leading</i>	Float oder Prozentwert	Zeilenabstand in Benutzerkoordinaten oder als prozentualer Anteil der Schriftgröße. Standardwert: 100%.
<i>parindent</i>	Float oder Prozentwert	Legt den linken Einzug der ersten Zeile eines Absatzes fest <sup>1</sup> . Der Wert wird zu <i>leftindent</i> addiert. Wird diese Option innerhalb der Zeile angegeben, so wirkt sie wie ein Tabulator. Standardwert: <i>o</i> .
<i>rightindent</i> <i>leftindent</i>	Float oder Prozentwert	Bestimmt den rechten bzw. linken Einzug aller Textzeilen <sup>1</sup> . Wird <i>leftindent</i> innerhalb der Zeile angegeben und befindet sich die definierte Position links der aktuellen Textposition, so wird die Option für die aktuelle Zeile ignoriert. Standardwert: <i>o</i> .
<i>rotate</i>	Float	Dreht das Koordinatensystem, mit der linken unteren Ecke der Fitbox als Mittelpunkt. Der übergebene Wert legt den Drehwinkel in Grad fest. Text und Box werden gedreht. Die Drehung wird zurückgesetzt, nachdem der Text platziert wurde. Standardwert: <i>o</i>
<i>ruler</i> <sup>2</sup>	Liste aus Floats oder Prozentwerten	Liste der absoluten Tabulatorpositionen für <i>hortabmethod=ruler</i> <sup>1</sup> . Die Liste darf maximal 32 nicht-negative Einträge in aufsteigender Reihenfolge enthalten. Standardwert: Vielfache von <i>hortabsize</i> als Integers.
<i>tabalignment</i>	Liste aus Schlüsselwörtern	Ausrichtung für Tabulatoren. Jeder Listeneintrag definiert die Ausrichtung des entsprechenden Eintrags in der Option <i>ruler</i> . Standardwert: <i>left</i> . <i>center</i> Text wird mittig an der Tabulatorposition ausgerichtet. <i>decimal</i> Das erste <i>tabalignchar</i> -Zeichen wird linksbündig an der Tabulatorposition ausgerichtet. Ist kein <i>tabalignchar</i> -Zeichen vorhanden, wird rechtsbündig ausgerichtet. <i>left</i> Text wird linksbündig an der Tabulatorposition ausgerichtet. <i>right</i> Text wird rechtsbündig an der Tabulatorposition ausgerichtet.

Tabelle 6.5 Textfluss-spezifische Blockeigenschaften

<b>Schlüsselwort</b>	<b>Typ</b>	<b>Mögliche Werte und Erklärung</b>
<i>verticalalign</i>	Schlüsselwort	Vertikale Ausrichtung des Texts in der Fitbox; die Optionen <i>firstlinedist</i> und <i>lastlinedist</i> werden entsprechend berücksichtigt. Standardwert: <i>top</i> . <i>top</i> Die Formatierung beginnt in der ersten Zeile und setzt sich nach unten fort. Füllt der Text die Fitbox nicht vollständig aus, bleibt Weißraum unter dem Text. <i>center</i> Der Text wird vertikal in der Fitbox zentriert. Füllt der Text die Fitbox nicht vollständig aus, bleibt Weißraum über und unter dem Text. <i>bottom</i> Die Formatierung beginnt in der letzten Zeile und setzt sich nach oben fort. Füllt der Text die Fitbox nicht vollständig aus, bleibt Weißraum über dem Text. <i>justify</i> Der Text wird am oberen und unteren Rand der Fitbox ausgerichtet. Dazu wird der Zeilenabstand bis zur durch <i>linespreadlimit</i> festgelegten Grenze erhöht. Die Höhe der ersten Zeile wird nur bei <i>firstlinedist=leading</i> vergrößert.
<b>Optionen zur Steuerung des Algorithmus für den Zeilenumbruch</b>		
<i>adjust-method</i>	Schlüsselwort	Methode zur Anpassung von Textteilen, die nach einer Vergrößerung oder Verkleinerung des Wortabstandes, der den in den Optionen <i>minspacing</i> und <i>maxspacing</i> definierten Grenzwerten unterliegt, nicht mehr in die Zeile passen. Standardwert: <i>auto</i> . <i>auto</i> Folgende Methoden werden in der angeführten Reihenfolge angewandt: <i>shrink</i> , <i>spread</i> , <i>nofit</i> , <i>split</i> . <i>clip</i> Wie <i>nofit</i> , nur dass der längere Teil am rechten Rand der Fitbox (unter Berücksichtigung der Option <i>rightindent</i> ) abgeschnitten wird. <i>nofit</i> Das letzte Wort wird in die nächste Zeile verschoben, sofern die verbleibende (kurze) Zeile nicht kürzer als der in der Option <i>nofitlimit</i> festgelegte Prozentwert ist. Auch Absätze im Blocksatz sehen bei dieser Methode leicht ausgefranst aus. <i>shrink</i> Passt ein Wort nicht in die Zeile, wird der Text so lange gestaucht, bis das Wort hineinpasst, sofern die Option <i>shrinklimit</i> dies zulässt. Anderenfalls kommt die Methode <i>nofit</i> zum Einsatz. <i>split</i> Das letzte Wort wird nicht in die nächste Zeile verschoben, sondern zwangsweise getrennt. Bei Textfonts (aber nicht bei Symbolfonts) wird ein Trennzeichen eingefügt. <i>spread</i> Das letzte Wort wird in die nächste Zeile verschoben. Der Rest der (kurzen) Zeile wird im Blocksatz ausgerichtet, indem der Zeichenabstand innerhalb der Wörter vergrößert wird, sofern die Option <i>spreadlimit</i> dies zulässt. Kann kein Blocksatz erreicht werden, kommt die Methode <i>nofit</i> zum Einsatz.
<i>linespreadlimit</i>	Float oder Prozentwert	(Nur für <i>verticalalign=justify</i> ) Größter Wert in Benutzerkoordinaten oder als Prozentsatz des Zeilenabstands, um den der Zeilenabstand bei vertikaler Ausrichtung erhöht wird. Standardwert: 200%.
<i>maxlines</i>	Integer oder Schlüsselwort	Maximale Anzahl der Zeilen in der Fitbox oder das Schlüsselwort <i>auto</i> , bei dem möglichst viele Zeilen in der Fitbox platziert werden. Nach der Platzierung der maximalen Anzahl von Zeilen gibt <i>PDF_fit_textflow()</i> den String <i>_boxfull</i> zurück.
<i>maxspacing</i> <i>minspacing</i>	Float oder Prozentwert	Bestimmt den maximalen bzw. minimalen Abstand zwischen Wörtern (in Benutzerkoordinaten oder als prozentualer Anteil der Breite eines Leerzeichens). Der berechnete Wortabstand wird durch hier übergebenen Werten begrenzt, aber der Wert der Option <i>wordspacing</i> wird noch addiert. Standardwerte: <i>minspacing=50%</i> , <i>maxspacing=500%</i> .
<i>nofitlimit</i>	Float oder Prozentwert	Minimale Zeilenlänge bei der Methode <i>nofit</i> (in Benutzerkoordinaten oder als prozentualer Anteil der Fitboxbreite). Standardwert: 75%.

Tabelle 6.5 Textfluss-spezifische Blockeigenschaften

Schlüsselwort	Typ	Mögliche Werte und Erklärung
<i>shrinklimit</i>	Prozentwert	Untere Grenze für das Stauchen von Text mit der Methode <i>shrink</i> . Der berechnete Stauchungsfaktor wird durch den hier übergebenen Wert begrenzt, aber noch mit dem Wert der Option <i>horizscaling</i> multipliziert. Standardwert: 85%.
<i>spreadlimit</i>	Float oder Prozentwert	Obere Grenze für den Abstand zwischen zwei Zeichen bei der Methode <i>spread</i> (in Benutzerkoordinaten oder als prozentualer Anteil der Schriftgröße); der berechnete Zeichenabstand wird durch den hier übergebenen Wert begrenzt, aber der Wert der Option <i>charspacing</i> wird noch addiert. Standardwert: 0.

1. In Benutzerkoordinaten oder als prozentualer Anteil an der Breite der Fitbox

2. Die Tabulatorpositionen werden auf der Registerkarte »Tabs« im Blockeigenschaften-Dialog festgelegt.

### 6.4.3 Rasterbildspezifische Eigenschaften

Für Blöcke vom Typ *Image* können neben den allgemeinen Eigenschaften einige rasterbildspezifische Eigenschaften definiert werden. Alle rasterbildspezifischen Eigenschaften lassen sich gemeinsam nutzen. Tabelle 6.6 gibt eine Übersicht.

Tabelle 6.6 Rasterbildspezifische Blockeigenschaften

Schlüsselwort	Typ	Mögliche Werte und Erklärung
<i>defaultimage</i>	String	Pfadname des Bildes, das verwendet wird, wenn vom Client kein Ersatzbild übergeben wird. Dateinamen sollten ohne absolute Pfadangabe verwendet werden und in der PPS-Client-Anwendung sollte mit der <i>SearchPath</i> -Funktionalität gearbeitet werden. Dies macht die Blockverarbeitung unabhängig von der Plattform und dateisystem-spezifischen Eigenheiten.
<i>dpi</i>	Float-Liste	Einer oder zwei Werte, die die gewünschte Bildauflösung in Pixel pro Zoll in horizontaler und vertikaler Richtung angeben. Ist der Wert gleich 0, wird die interne Bildauflösung verwendet. Ist diese nicht vorhanden, werden 72 dpi benutzt. Diese Eigenschaft wird ignoriert, wenn die Eigenschaft <i>fitmethod</i> mit einem der Schlüsselwörter <i>auto</i> , <i>meet</i> , <i>slice</i> oder <i>entire</i> übergeben wurde. Standardwert: 0.
<i>scale</i>	Float-Liste	Einer oder zwei Werte, die den oder die gewünschten Skalierungsfaktoren in horizontaler und vertikaler Richtung festlegen. Diese Eigenschaft wird ignoriert, wenn die Eigenschaft <i>fitmethod</i> mit einem der Schlüsselwörter <i>auto</i> , <i>meet</i> , <i>slice</i> oder <i>entire</i> übergeben wurde. Standardwert: 1.

## 6.4.4 PDF-spezifische Eigenschaften

Für Blöcke vom Typ *PDF* können neben den allgemeinen Eigenschaften einige PDF-spezifische Eigenschaften definiert werden. Alle PDF-spezifischen Eigenschaften lassen sich gemeinsam nutzen. Tabelle 6.7 gibt eine Übersicht.

Tabelle 6.7 PDF-spezifische Blockeigenschaften

<b>Schlüsselwort</b>	<b>Typ</b>	<b>Mögliche Werte und Erklärung</b>
<i>defaultpdf</i>	String	Pfadname des PDF-Dokuments, das verwendet wird, wenn vom Client kein Ersatzdokument übergeben wird. Dateinamen sollten ohne absolute Pfadangabe verwendet werden und in der PPS-Client-Anwendung sollte mit der SearchPath-Funktionalität gearbeitet werden. Dies macht die Blockverarbeitung unabhängig von der Plattform und dateisystem-spezifischen Eigenheiten.
<i>default-pdfpage</i>	Float	Nummer der Seite im Standard-PDF-Dokument. Standardwert: 1.
<i>scale</i>	Float-Liste	Einer oder zwei Werte, die den oder die gewünschten Skalierungsfaktoren in horizontaler und vertikaler Richtung festlegen. Diese Eigenschaft wird ignoriert, wenn die Eigenschaft <i>fitmethod</i> mit einem der Schlüsselwörter <i>auto</i> , <i>meet</i> , <i>slice</i> oder <i>entire</i> übergeben wurde. Standardwert: 1.
<i>pdusebox</i>	Schlüsselwort	(Mögliche Werte: <i>media</i> , <i>crop</i> , <i>bleed</i> , <i>trim</i> , <i>art</i> ) Verwendet die <i>MediaBox</i> , <i>CropBox</i> , <i>BleedBox</i> , <i>TrimBox</i> oder <i>ArtBox</i> der platzierten Seite, um deren Größe zu bestimmen (siehe Tabelle 8.44). Standardwert: <i>crop</i> .

## 6.4.5 Selbstdefinierte Eigenschaften

Selbstdefinierte Eigenschaften können zusätzlich zu den allgemeinen und typspezifischen Eigenschaften für Blöcke beliebigen Typs definiert werden. Selbstdefinierte Eigenschaften sind optional und lassen sich nicht gemeinsam nutzen. Tabelle 6.8 gibt eine Übersicht über die Namensregeln für selbstdefinierte Eigenschaften.

Table 6.8 Selbstdefinierte Blockeigenschaften

<b>Schlüsselwort</b>	<b>Typ</b>	<b>Mögliche Werte und Erklärung</b>
<i>beliebiger Name, der die drei Zeichen [ ] / nicht enthalten darf</i>	String, Name, Float, Float-Liste	Die Interpretation der Werte selbstdefinierter Eigenschaften liegt vollständig bei der Client-Applikation.

## 6.5 Abfragen von Blocknamen und -eigenschaften

Neben automatischer Blockverarbeitung unterstützt PDFlib einige Funktionen, mit denen sich Blocknamen auflisten sowie Standard- und selbstdefinierte Eigenschaften abfragen lassen.

**Ermitteln der Anzahl und Namen von Blöcken.** Die Namen und die Anzahl der Blöcke auf einer importierten Seite brauchen dem Clientcode nicht bekannt zu sein, da sie abgefragt werden können. Die folgende Anweisung gibt die Anzahl der Blöcke auf der Seite zurück:

```
blockcount = PDF_get_pdi_value(p, "vdp/blockcount", doc, page, 0);
```

Die folgende Anweisung gibt den Namen von Block Nummer 5 auf der Seite (die Zählung beginnt bei 0) oder einen Leerstring zurück, wenn kein entsprechender Block existiert (hat der Parameter oder die Option *pdiwarning* den Wert *true*, wird jedoch eine Exception ausgelöst):

```
blockname = PDF_get_pdi_parameter(p, "vdp/Blocks[5]/Name", doc, page, 0, &len);
```

Der zurückgegebene Blockname kann im weiteren Verlauf zur Abfrage von Blockeigenschaften oder zum Füllen des Blocks mit Text, Bildern oder PDF-Inhalt verwendet werden.

In der Syntax für den Pfad zur Addressierung der Blockeigenschaft sind folgende Ausdrücke gleichbedeutend, wobei davon ausgegangen wird, dass der Block mit der Nummer *<nummer>* seine Eigenschaft *Name* auf *<blockname>* gesetzt hat:

```
Blocks[<nummer>]/  
Blocks/<blockname>/
```

**Ermitteln von Blockkoordinaten.** Die beiden Koordinatenpaare (*llx*, *lly*) und (*urx*, *ury*), die die linke untere und die rechte obere Ecke eines Blocks namens *foo* beschreiben, lassen sich wie folgt abfragen:

```
llx = PDF_get_pdi_value(p, "vdp/Blocks/foo/Rect[0]", doc, page, 0);  
lly = PDF_get_pdi_value(p, "vdp/Blocks/foo/Rect[1]", doc, page, 0);  
urx = PDF_get_pdi_value(p, "vdp/Blocks/foo/Rect[2]", doc, page, 0);  
ury = PDF_get_pdi_value(p, "vdp/Blocks/foo/Rect[3]", doc, page, 0);
```

Beachten Sie, dass diese Koordinaten im Standard-Benutzerkoordinatensystem (mit dem Ursprung in der linken unteren Ecke, eventuell modifiziert durch die CropBox der Seite) übergeben werden, wohingegen das Block-Plugin die Koordinaten gemäß des Koordinatensystems der Acrobat-Benutzeroberfläche mit dem Ursprung in der linken oberen Ecke der Seite anzeigt.

Beachten Sie zudem, dass der Parameter *topdown* bei der Abfrage der Blockkoordinaten nicht berücksichtigt wird.

**Abfrage von selbstdefinierten Eigenschaften.** Selbstdefinierte Eigenschaften lassen sich wie in folgendem Beispiel abfragen, in dem die Eigenschaft *zipcode* von einem Block namens *b1* abgefragt wird.

```
zip = PDF_get_pdi_parameter(p, "vdp/Blocks/b1/Custom/zipcode", doc, page, 0, &len);
```

**Namensraum für selbstdefinierte Eigenschaften.** Zur Vermeidung von Namenskonflikten beim Austausch von PDF-Dokumenten aus verschiedenen Quellen sollten die Namen selbstdefinierter Eigenschaften aus einem firmeneigenen Präfix, gefolgt von einem Strichpunkt ':' und dem eigentlichen Eigenschaftsnamen, bestehen. Als firmenspezifisches Präfix empfehlen wir den jeweiligen Internet-Domainnamen. Die Property-Namen des Unternehmens ACME sähen dann zum Beispiel wie folgt aus:

```
acme.com:digits  
acme.com:refnumber
```

Da Standard- und selbstdefinierte Eigenschaften im Block unterschiedlich gespeichert werden, können Standardnamen (wie in Abschnitt 6.4, »Standardeigenschaften zur automatischen Verarbeitung«, Seite 175, definiert) nicht mit selbstdefinierten Namen in Konflikt geraten.

## 6.6 Spezifikation für PDFlib-Blöcke

Die Syntax für PDFlib-Blöcke ist vollständig kompatibel zur PDF-Referenz, die einen Erweiterungsmechanismus definiert, mit dem eine Applikation private Daten als Anhang an die Datenstrukturen einer PDF-Seite speichern kann. Der folgende Abschnitt beschreibt die Syntax der PDFlib-Blöcke im Detail. Dies kann für solche Anwender von Nutzen sein, die PDFlib-Blöcke nicht mit dem Block-Plugin, sondern auf andere Art erzeugen wollen. Anwender, die mit dem Plugin arbeiten, können diesen Abschnitt gestrost überspringen.

### 6.6.1 PDF-Objektstruktur für PDFlib-Blöcke

Das Page-Dictionary enthält einen Eintrag */PiecelInfo*, der als Wert ein weiteres Dictionary enthält. Dieses Dictionary enthält den Key */PDFlib*, der als Wert wiederum ein Dictionary mit Applikationsdaten enthält. Das Applikationsdaten-Dictionary enthält die beiden in Tabelle 6.9 aufgeführten Standardkeys.

Tabelle 6.9 Einträge in einem PDFlib-Applikationsdaten-Dictionary

Key	Typ	Wert
<i>LastModified</i>	Datum	(Erforderlich) Datum und Zeit der Erstellung oder letzten Änderung der Blöcke auf der Seite.
<i>Private</i>	Dictionary	(Erforderlich) Blockliste gemäß Tabelle 6.10

Eine Blockliste ist ein Dictionary mit allgemeinen Information zur Blockverarbeitung sowie einer Liste aller Blöcke auf der Seite. Tabelle 6.10 enthält alle Keys in einem Blocklisten-Dictionary.

Tabelle 6.10 Einträge in einem Blocklisten-Dictionary

Key	Typ	Wert
<i>Version</i>	Zahl	(Erforderlich) Die Versionsnummer der Blockspezifikation, gemäß der die Datei erstellt wurde. Dieses Dokument beschreibt Version 5 der Blockspezifikation.
<i>Blocks</i>	Dictionary	(Erforderlich) Jeder Schlüssel ist ein Namensobjekt mit dem Namen eines Blocks; der zugehörige Wert ist das Block-Dictionary des Blocks (siehe Tabelle 6.12). Der Schlüssel <i>/Name</i> im Block-Dictionary muss identisch zum Namen des Blocks in diesem Dictionary sein.
<i>PluginVersion</i>	String	(Erforderlich, falls der Schlüssel <i>pdfmark</i> nicht vorhanden ist <sup>1</sup> ) Ein String, der die Versionsnummer des PDFlib Block-Plugins enthält, mit dem die Blöcke erstellt wurden.
<i>pdfmark</i>	Boolean	(Erforderlich, falls der Schlüssel <i>PluginVersion</i> nicht vorhanden ist <sup>1</sup> ) Muss den Wert <i>true</i> haben, falls die Blockliste mit Hilfe von <i>pdfmarks</i> erstellt wurde.

1. Genau einer der Keys *PluginVersion* und *pdfmark* muss vorhanden sein.

**Datentypen von Blockeigenschaften.** Für Blockeigenschaften werden mit Ausnahme von Handles und Aktionslisten dieselben Datentypen wie für Optionslisten unterstützt (siehe Abschnitt 3.1.4, »Optionslisten«, Seite 51). Tabelle 6.11 zeigt, wie diese Typen auf PDF-Datentypen abgebildet werden.

Tabelle 6.11 Datentypen für Blockeigenschaften

Blocktyp	PDF-Typ	Anmerkung
Boolean	Boolean	
String	String	
Schlüsselwort	String	Schlüsselwörter dürfen nur in der Liste der Schlüsselwörter, die von einer bestimmten Eigenschaft unterstützt werden, übergeben werden. Andernfalls führt dies zu einem Fehler.
Float, Integer	Boolean	Optionslisten akzeptieren Punkt und Komma als Dezimalzeichen, bei PDF-Zahlen dagegen ist nur ein Punkt erlaubt.
Prozentwert	Array mit zwei Elementen	Das erste Arrayelement enthält die Zahl, das zweite Element einen String mit dem Prozentzeichen.
Farbe	Array mit zwei Elementen	Das erste Arrayelement legt einen Farbraum fest und das zweite einen Farbwert. Für das erste Arrayelement werden folgende Einträge unterstützt: <i>/DeviceGray</i> Das zweite Element ist ein einzelner Graustufenwert. <i>/DeviceRGB</i> Das zweite Element ist ein Array mit drei RGB-Werten. <i>/DeviceCMYK</i> Das zweite Element ist ein Array mit vier CMYK-Werten. <i>[/Separation/spotname]</i> Das erste Element ist ein Array mit dem Schlüsselwort <i>/Separation</i> und dem Farbnamen. Das zweite Element ist ein Wert für den Farbauftrag. <i>[/Lab]</i> Das erste Element ist ein Array mit dem Schlüsselwort <i>/Lab</i> . Das zweite Element ist ein Array aus drei Lab-Werten.  Um keine Farbe festzulegen, muss die entsprechende Eigenschaft weggelassen werden.

**Schlüssel in Block-Dictionaries.** Block-Dictionaries können die Schlüssel in Tabelle 6.12 enthalten. Abhängig vom Wert des Keys */Subtype* dürfen nur Keys aus einer der Gruppen Text, Image oder PDF vorhanden sein (siehe Tabelle 6.3).

Tabelle 6.12 Einträge in einem Block-Dictionary

Key	Typ	Wert
Allgemeine Eigenschaften		(Manche Keys sind erforderlich) Allgemeine Blockeigenschaften gemäß Tabelle 6.3
Text-Eigenschaften		(Optional) Text- und textfluss-spezifische Blockeigenschaften gemäß Tabelle 6.4 und Tabelle 6.5
Image-Eigenschaften		(Optional) Rasterbildspezifische Blockeigenschaften gemäß Tabelle 6.6
PDF-Eigenschaften		(Optional) PDF-spezifische Blockeigenschaften gemäß Tabelle 6.7
Custom	Dictionary	(Optional) Dictionary mit Schlüssel/Wert-Paaren für selbstdefinierte Eigenschaften gemäß Tabelle 6.8
Internal	Dictionary	(Optional) Dieser Schlüssel ist für internen Gebrauch reserviert; Anwendungen sollten keinerlei Annahmen über sein Vorhandensein oder bestimmte Inhalte machen. Derzeit wird dieser Schlüssel zur Verwaltung der Beziehung zwischen konvertierten Formularfeldern und den zugehörigen Blöcken genutzt.

**Beispiel.** Das folgende Fragment zeigt den PDF-Code für zwei Blöcke, einen Textblock namens *job\_title* und einen Image-Block namens *logo*. Der Textblock enthält eine selbstdefinierte Eigenschaft namens *format*:

```

<<
  /Contents 12 0 R
  /Type /Page
  /Parent 1 0 R
  /MediaBox [ 0 0 595 842 ]
  /PieceInfo << /PDFlib 13 0 R >>
>>

13 0 obj
<<
  /Private <<
    /Blocks <<
      /job_title 14 0 R
      /logo 15 0 R
    >>
    /Version 5
    /PluginVersion (2.1.0)
  >>
  /LastModified (D:20050813200730)
>>
endobj

14 0 obj
<<
  /Type /Block
  /Rect [ 70 740 200 800 ]
  /Name /job_title
  /Subtype /Text
  /fitmethod /auto
  /fontname (Helvetica)
  /fontsize 12
  /Custom << /format 5 >>
>>
endobj

15 0 obj
<<
  /Type /Block
  /Rect [ 250 700 400 800 ]
  /Name /logo
  /Subtype /Image
  /fitmethod /auto
>>

```

## 6.6.2 Erzeugen von PDFlib-Blöcken mit pdfmarks

Alternativ zur Erzeugung von PDFlib-Blöcken mit dem Plugin können Sie Blöcke auch durch Einfügen von pdfmark-Anweisungen in einen PostScript-Datenstrom und anschließendes Destillieren erzeugen. Einzelheiten zum pdfmark-Operator werden in der Acrobat-Dokumentation beschrieben. Das folgende Fragment zeigt pdfmark-Operatoren, die die Blockdefinition im vorhergehenden Abschnitt erzeugen:

```

% ----- Vorbereitung für die Blöcke der Seite -----
[/_objdef {B1} /type /dict /OBJ pdfmark          % Blocks-Dictionary

[ {ThisPage} <<
  /PieceInfo <<
    /PDFlib <<

```

```

        /LastModified (D:20050813200730)
        /Private <<
            /Version 5
            /pdfmark true
            /Blocks {B1}
        >>
    >>
>> /PUT pdfmark

% ----- Textblock -----
[ {B1} <<
    /job_title <<
        /Type /Block
        /Name /job_title
        /Subtype /Text
        /Rect [ 70 740 200 800 ]
        /fitmethod /auto
        /fontsize 12
        /fontname (Helvetica)
        /Custom << /format 5 >>
    >>
>> /PUT pdfmark

% ----- Image-Block -----
[ {B1} <<
    /logo <<
        /Type /Block
        /Name /logo
        /Subtype /Image
        /Rect [ 250 700 400 800 ]
        /fitmethod /auto
    >>
>> /PUT pdfmark

```