

# 1 Der PDFMark Operator

## 1.1 Programmieren von PostScript-Dateien mit PDF-Funktionen

Acrobat Distiller konvertiert PostScript-Dateien in das Portable Document Format (PDF). Dieses Format unterstützt spezielle Funktionen wie Notizen, Verknüpfungen, Lesezeichen und Seitenbeschnitt. Normalerweise sind diese Funktionen nicht Bestandteil einer PostScript-Datei. Im folgenden wird der Gebrauch und die Syntax des PDFMark Operators beschrieben, welcher im PostScript Level 2 Interpreter von Acrobat Distiller ab Version 2.0 enthalten ist. Neuere Programmversionen unterstützen zusätzliche Funktionen.

### 1.1.1 Syntax

Für jede der Eingangs beschriebenen Funktionen ist ein Eintrag in der PostScript-Datei notwendig. Dieser Eintrag benutzt den neuen **pdfmark** Operator. Er besteht aus einem Mark Objekt, verschiedenen Key-Value Paaren, und einem Namen. Der **pdfmark** Operator hinterläßt keine Werte auf dem Stack. Jedesmal wenn der **pdfmark** Operator in der PostScript-Datei aufgerufen wird, setzt er eine Markierung.

*Ausgabe von  
PostScript-Dateien mit  
pdfmark auf  
PostScriptdruckern.*

Damit PostScript-Dateien, die den **pdfmark** Operator enthalten auch auf Druckern oder anderen Ausgabegeräten verwendet werden können, sollte der Prolog der PostScript-Datei folgenden Programmcode enthalten:

```
systemdict /pdfmark known not  
{userdict /pdfmark systemdict /cleartomark get put}  
if
```

Ist der Eintrag **pdfmark** nicht im **systemdict** vorhanden, so wird folgendes Key-Value Paar im **userdict** angelegt:  
KEY = /pdfmark VALUE = Wert von /cleartomark

Dem Namen **pdfmark** wird also der Programmcode von **cleartomark** zugeordnet, welcher den Stack bis einschließlich der Marke freigibt.

Dieser Abschnitt behandelt die verschiedenen Verwendungsmöglichkeiten des **pdfmark** Operators. Die Definition jedes im folgenden verwendeten Key-Value Paares folgt im Abschnitt 1.1.2.

### 1.1.1.1 Notizen

Notizen verwenden **pdfmark** zusammen mit dem Namen **ANN**. Die Syntax für eine Notiz:

```
[ /Contents string
  /Rect [ulx uly orx ory]
  /SrcPg pagenum
  /Open boolean
  /Color colorspec
  /Title string
  /ModDate datestring
/ANN pdfmark
```

Die Werte **Contents** und **Rect** müssen vorhanden sein. Alle anderen Werte sind optional.

Der Eintrag für eine Notiz muß – innerhalb der PostScript-Datei – auf der Seite erfolgen, auf der sie erscheinen soll. Wird der Key **SrcPg** verwendet, kann die Notiz überall in der PostScript-Datei eingefügt werden.

### 1.1.1.2 Verknüpfungen

Verknüpfungen verwenden **pdfmark** zusammen mit dem Namen **LNK**. Die Syntax für eine Verknüpfung:

```
[ /Action action
  /Dest nameddest
  /Page pagenum
  /View [destination]
  /File string
  /MacFile string
  /DOSFile string
  /UnixFile string
  /ID [string string]
  /WinFile string
```

```
/Dir string
/Op string
/Params string
/Unix string
/Rect [ulx uly orx ory]
/SrcPg pagenum
/Border [bx by c [d]]
/Color colorspec
/Title string
/ModDate datestring
/LNK pdfmark
```

Die **Action**, **SrcPg**, **Border**, **Color** und **Titel** Keys sind optional. Die **Dest**, **Page**, **View**, **File**, **MacFile**, **DOSFile**, **UnixFile**, **ID**, **WinFile**, **Dir**, **Op**, **Params** und **Unix** Keys können, in Abhängigkeit der Verwendung des **Action** Wertes, optional sein.

Der Eintrag für eine Verknüpfung muß – innerhalb der PostScript-Datei – auf der Seite erfolgen, von der aus sie aufgerufen werden soll. Wird der Key **SrcPg** verwendet, kann die Verknüpfung überall in der PostScript-Datei eingefügt werden.

### 1.1.1.3 Lesezeichen

Lesezeichen verwenden **pdfmark** zusammen mit dem Namen **OUT**. Die Syntax für ein Lesezeichen:

```
[ /Action action
  /Dest nameddest
  /Page pagenum
  /View [destination]
  /File string
  /MacFile string
  /DOSFile string
  /UnixFile string
  /ID [string string]
  /WinFile string
  /Dir string
  /Op string
  /Params string
  /Unix string
  /Title string
  /Count int
/OUT pdfmark
```

Die **Action** and **Count** keys sind optional. Die **Dest**, **Page**, **View**, **File**, **MacFile**, **DOSFile**, **UnixFile**, **ID**, **WinFile**, **Dir**, **Op**, **Params** und **Unix** Keys können, in Abhängigkeit der Verwendung des **Action** Wertes, optional sein.

Einträge für Lesezeichen können überall in der PostScript-Datei erfolgen.

#### 1.1.1.4 Artikel (Ab Acrobat 2.0)

Artikel bestehen aus Aneinanderreihungen rechteckiger Bereiche. Dem Leser wird dadurch das Lesen eines Dokumentes erleichtert.

Artikel verwenden **pdfmark** zusammen mit dem Namen **ARTICLE**. Die Syntax für einen Artikel:

```
[ /Title string
  /Subject string
  /Author string
  /Keywords string
  /Rect [ulx uly orx ory]
  /Page pagenum
/ARTICLE pdfmark
```

Die **Subject**, **Author**, **Keywords** und **Page** Keys sind optional. Ein Eintrag für einen Artikel kann überall in der PostScript-Datei erfolgen. Der Artikel wird durch den **Title** definiert. Besteht der Artikel aus mehreren „Beads“ (Reihe von definierten rechteckigen Bereichen), so ist deren Reihenfolge im PDF-Dokument, mit der in der PostScript-Datei identisch.

#### 1.1.1.5 Benannte Ziele (Ab Acrobat 2.0)

Der Sprung an ein bestimmtes Ziel wird durch die **Page** und **View** Keys erreicht. Das Ziel kann jedoch auch durch den **Dest** Key definiert werden. Dies ist dann überall in der PostScript-Datei oder in einer anderen Datei möglich. Ein benanntes Ziel verwendet **pdfmark** zusammen mit dem Namen **DEST**. Die Syntax:

```
[ /Dest name
  /Page pagenum
  /View [destination]
/DEST pdfmark
```

Die **Page** und **View** Keys sind optional. Weitere Hinweise zu **Dest** im Abschnitt 2.1.2.

#### 1.1.1.6 Pass-through PostScript (Ab Acrobat 2.0)

*Tip: Somit ist es ab der Version 2.0 von Adobe Acrobat möglich, benutzerdefinierte Rasterfunktionen zu verwenden.*

„Pass-through PostScript“ verwendet **pdfmark** zusammen mit dem Namen **PS**. Dadurch besteht die Möglichkeit PostScript-Blöcke in die Datei einzufügen. Diese Blöcke werden von Acrobat Distiller und allen Acrobat Viewern ignoriert. Beim Druck auf einen PostScript-Drucker werden sie jedoch ausgegeben. Die Syntax für einen PostScript-Block:

```
[ /DataSource string|file
  /Level1 string|file
/PS pdfmark
```

Der **Level1** Key ist optional. Dieser Marker muß in der PostScript-Datei an der Stelle eingefügt werden, an der der PostScript-Code während des druckens interpretiert werden soll.

#### 1.1.1.7 Seitenbeschnitt

Seitenbeschnitt verwendet **pdfmark** zusammen mit dem Namen **PAGES** und **PAGE**. Die Syntax für den Seitenbeschnitt:

```
[/CropBox [ulx uly orx ory] /PAGES pdfmark
```

Dieser Marker kann zwar überall in der PostScript-Datei platziert werden, jedoch sollte er vor jedem weiteren Marker der Seite stehen. Es ist üblich ihn im Setup-Teil der PostScript-Datei – also zwischen **%%BeginSetup** und **%%EndSetup** – anzuführen. Ist der Distiller so eingestellt, daß er automatisch Thumbnails generiert, so werden diese falsch erstellt, wenn sich der Seitenbeschnitt-Marker am Ende der Datei befindet.

**Pages** gilt für alle Seiten eines Dokumentes. Soll diese Vorgabe für eine bestimmte Seite geändert werden, ist folgende Syntax für den **PAGE** Marker anzuwenden:

```
[/CropBox [ulx uly orx ory] /PAGE pdfmark
```

Dieser Eintrag muß vor dem **showpage** Operator der Seite erfolgen.

#### 1.1.1.8 Info und Catalog (Ab Acrobat 2.0)

Informationen für das Info und Catalog Dictionary werden mit **pdfmark** zusammen mit den Namen **DOCINFO** und **DOCVIEW** erstellt. Die Syntax um Informationen für das Catalog Dictionary zu definieren:


```
[ /Action action
  /Dest nameddest
  /Page pagenum
  /View [destination]
  /File string
  /MacFile string
  /DOSFile string
  /UnixFile string
  /ID [string string]
  /WinFile string
  /Dir string
  /Op string
  /Params string
  /Unix string
  /PageMode name
/DOCVIEW pdfmark
```

Die **Action** und **PageMode** Keys sind optional. Die **Dest**, **Page**, **View**, **File**, **MacFile**, **DosFile**, **UnixFile**, **ID**, **WinFile**, **Dir**, **Op**, **Params** und **Unix** Keys können, in Abhängigkeit des **Action** Wertes, optional sein. Weitere Hinweise zu den Key-Value Definitionen folgen in Abschnitt 1.1.2.

Ein Beispiel für die Syntax um Informationen für das Info Dictionary zu definieren:

```
[ /Author string
  /CreationDate string
  /Creator string
  /Producer string
  /Title string
  /Subject string
  /Keywords string
  /ModDate string
/DOCINFO pdfmark
```

### 1.1.2 Definitionen der Key-Value Paare


Die folgende Tabelle definiert die Key-Value Paare, welche mit dem **pdfmark** Operator verwendet werden. Für Keys, die ab Adobe Acrobat 2.0 verfügbar sind, wird das Symbol  als Kennzeichnung verwendet.

Key	Typ	Bedeutung
<b>Rect</b>	array	Definiert – in Point – die Größe und Lage eines Rechtecks, relativ zur unteren linken Ecke einer Seite. <b>Rect</b> ist ein Array, bestehend aus vier Zahlen, [ulx, uly, orx, ory]; ulx steht für x unten links, uly für y unten links, orx für x oben rechts und ory für y oben rechts – alles Koordinaten im User Space. Betroffen sind geöffnete Notizen oder Verknüpfungs-Felder.
<b>Open</b>	boolean	Gibt an, ob eine Notiz geöffnet oder geschlossen ist. Fehlt dieser Key, so ist die Notiz geschlossen.
<b>Contents</b>	string	Enthält den Text-String für eine Notiz. Die maximale Länge des Strings ist 65.535 Zeichen. Benützt wird der PDFDocEncoding-Vektor (Vgl. Portable Document Format Reference Manual). Ob die Zeichen verfügbar sind hängt jedoch auch davon ab, ob sie im Zeichensatz des Betriebssystems vorhanden sind.
<b>Border</b>	array	Bestimmt das Aussehen der Umrandung einer Verknüpfung. <b>Border</b> ist ein Array, bestehend aus drei Zahlen (in Point, im User Space), [bx by c]; bx steht für den horizontalen Eckenradius, by für den vertikalen Eckenradius und c für die Strichdicke der Umrandung. Die Standardwerte für den optionalen <b>Border</b> Key sind 0, 0 und 1. Ist die Strichdicke 0, so wird keine Umrandung dargestellt. Wahlweise kann dem c Key ein Dash Array folgen, z.B. [bx by c [d]]. Der Dash Array bestimmt die Strichlänge einer gestrichelten Umrandung. <i>Hinweis: „Das PDF 1.0 Reference Manual gibt an, daß die ersten beiden Zahlen die horizontalen und vertikalen Eckenradien beschreiben; 1.0 Viewer ignorieren diese Information. Diese Information wird ebenso von 2.0 und zukünftigen Viewern ignoriert.“</i> [Adobe Systems, <i>Updates to the Portable Document Format Reference Manual</i> , Tech. Note #5156, April 1995]

**CropBox** array Definiert – in Point – die sichtbare Größe einer Seite, relativ zur unteren linken Ecke. **CropBox** ist ein Array, bestehend aus vier Zahlen, [ulx, uly, orx, ory]; ulx steht für x unten links, uly für y unten links, orx für x oben rechts und ory für y oben rechts – alles Koordinaten im User Space. Die kleinste Seitengröße beträgt 1 x 1 inch (72 x 72 Dots im Koordinatensystem des User Space), die maximale Seitengröße ist 45 x 45 inch (3240 x 3240 Dots).

### PageMode

name Bestimmt die Gestalt des Dokumentes beim Öffnen. Zulässige Werte:  
**UseNone** – Öffnet das Dokument ohne Lesezeichen und Thumbnails.  
**UseOutlines** – Öffnet das Dokument mit sichtbaren Lesezeichen.  
**UseThumbs** – Öffnet das Dokument mit sichtbaren Thumbnails.  
**FullScreen** – Öffnet das Dokument im Vollbild-Modus, ohne Window.

**SrcPg**  integer Bestimmt die Ausgangsseite, auf der eine Verknüpfung oder eine Notiz wirksam wird. Die laufende Nummerierung der Seiten beginnt – gemäß der Adobe Dokumenten-Strukturierungsvereinbarung: %%Page – bei 1, nicht 0.


**Count** integer Bestimmt die Anzahl und Anzeige von untergeordneten Lesezeichen. Ein Lesezeichen kann sich in mehrere Unterlesezeichen aufgliedern (sog. Outline Tree). Bei der Benutzung von **pdfmark** zusammen mit dem **Count** Key sind einige Regeln zu beachten:

1. In der PostScript-Datei müssen Einträge für Lesezeichen aufeinanderfolgend aufgeführt werden.
2. Ein Lesezeichen ohne „Unterlesezeichen“ benötigt keinen **Count** Key.
3. Ein Lesezeichen mit „Unterlesezeichen“ benötigt den **Count** Key. Sein Wert gibt die Anzahl der untergeordneten Lesezeichen an und zwar inklusive der Anzahl der „Unter-Unterlesezeichen“. Ist der Wert positiv, so ist das Ausgangslesezeichen geöffnet, ist der Wert negativ, so ist es geschlossen.

**Title**            string    Enthält die Bezeichnung für ein Lesezeichen oder den Titel eines Artikels. Optional kann auch ein Titel für eine Verknüpfung oder eine Notiz angegeben werden. Der **Title** String kann maximal 65.535 Zeichen enthalten. Aus Lesbarkeitsgründen sollten jedoch nicht mehr als 32 Zeichen verwendet werden. Diese benutzen den PDFDocEncoding-Vektor. Bei Artikeln dient der Titel zur Erkennung des Artikels, falls er durch einen Eintrag angesprochen wird.

#### **DataSource**

string|file    Bestimmt, welchen Ursprung der PostScript-Quellcode für Pass-through PostScript hat. Der Wert muß entweder ein String oder ein Datei-Objekt sein. Beide sind in runde Klammern zu setzen. Da PostScript den Backslash als Steuerzeichen interpretiert, muß der DOS Zugriffspfad an der Stelle des Backslash noch einen weiteren enthalten, z.B.:  
C:\\ACROBAT\\test.pdf


**Color**     array    Gibt an, welche Farbe die Umrandung der Ursprung einer Verknüpfung, der Hintergrund eines Notizsymbols oder die Titelleiste einer geöffneten Notiz erhält. Der Wert ist ein Array aus Zahlen von 0 bis 1. Die Länge des Arrays bestimmt den Farbraum, der verwendet wird. Eine Länge von 1 entspricht **DeviceGray**, eine Länge von 3 entspricht **DeviceRGB**, eine Länge von 4 entspricht **DeviceCMYK**. (Hinweise zu diesen Farbräumen vgl. Portable Document Format Reference Manual, Abschnitt 6.8.5)

#### **ModDate**

string    Enthält das Datum der letzten Änderung einer Notiz oder einer Verknüpfung. Es wird jedes Format akzeptiert, jedoch sollte man das bevorzugte Format für Acrobat Viewer verwenden:

(D:YYYYMMDDHHmmSS)

Das Präfix D: ist zwar optional, wird jedoch benötigt damit ein String als Datumsformat erkannt wird (Weiteres siehe Kapitel 5.4.3, Info Dictionary). YYYY steht für das Jahr, MM für den Monat (01-12), DD für den Tag (01-31), HH für die Stunde (00-23), mm für die Minuten (00-59) und SS für die Sekunden (00-59). Alle Felder nach dem Jahr sind optional. *Hinweis: Acrobat 1.0 Viewer interpretieren das Datumsformat als „normalen“ String.*

**Action**  name Bestimmt den Typ einer Aktion für Verknüpfungen, Lesezeichen und Dokumente. Es werden vier mögliche Aktionen unterstützt:

1. **GoTo** – Springt innerhalb des aktuellen Dokumentes zu einer anderen Seite, mit bestimmtem Zoom-Faktor.
2. **GoToR** – Öffnet ein anderes PDF-Dokument an einer bestimmten Seite und einem bestimmten Zoom-Faktor.
3. **Launch** – Öffnet eine andere Anwendung. (Dies kann auch ein PDF-Dokument sein).
4. **Article** – Sprung zu einem Artikel, sowohl im aktuellen als auch in einem anderen PDF-Dokument.

Um die Aktion genau zu definieren, werden weitere Keys benötigt:

1. Die **Page**, **View** oder **Dest** Keys bestimmen das Ziel einer Aktion innerhalb eines PDF-Dokumentes. Die **GoTo** und **GoToR** Aktionen können diese Keys verwenden.
2. Die **Article** Aktion verwendet nur den **Dest** Key, dieser ist jedoch verbindlich.
3. Die **File**, **MacFile**, **DOSFile**, **UnixFile** und **ID** Keys legen eine externe Datei für eine Aktion fest. Aktionen mit **GoToR**, **Launch** und **Article** können diese Keys benutzen.
4. Die **Unix**, **WinFile**, **Dir**, **Op** und **Params** Keys sind nur zusammen mit der **Launch** Aktion zu verwenden. Sie sind alle optional.






**Dest** 

destination Bestimmt das Ziel einer **Article** Aktion oder einer benannten Verknüpfung für jede Aktion. Der Wert einer benannten Verknüpfung muß durch den **DEST** Operator als Name definiert worden sein. Ist das Ziel ein Artikel, kann der Wert eine Integer-Zahl, welche den Artikel-Index bestimmt (Beginn der Zählung bei Integer = 0), oder ein String, der mit dem Titel des Artikels identisch ist, sein.

**Page**

integer | name Definiert die Zielseite für **GoTo** und **GoToR** Aktionen sowie für Verknüpfungen und Artikel. Bei Verknüpfungen und Artikeln sind die Namen **Next** und **Prev** gültige Werte für




eine Zielseite. Der **Page** Key sollte nicht verwendet werden, wenn die Zielseite einer Verknüpfung die aktuelle Seite ist. Ist der Wert des **Page** Keys 0, so existiert kein Ziel.

<b>View</b>	array	<p><b>View</b> ist ein Array, der das Ziel einer Verknüpfung oder eines Lesezeichens z.B. bei <b>GoTo</b> und <b>GoToR</b> Aktionen definiert. Der Inhalt des <b>View</b> Array wird in Abschnitt 1.1.3 beschrieben.</p> <p>Die Informationen des <b>Page</b> und <b>View</b> Keys werden von Acrobat Distiller zu einem anderen Key (<b>Dest</b>) kombiniert und in das PDF-Dokument eingefügt.</p>
<b>File</b> 	string	<p>Definiert den systemunabhängigen Pfad einer PDF-Datei für <b>GoToR</b> und <b>Article</b> Aktionen oder einer <b>Launch</b> Aktion. Ein Beispiel für einen systemunabhängigen Pfad: Der MS-DOS-Pfad C:\ACROBAT\Test.pdf entspricht dann /C/ACROBAT/Test.pdf</p>
<b>MacFile</b> 	string	<p>Bestimmt den Macintosh Dateinamen im PDF-Format für Pfadnamen bei <b>GoToR</b> und <b>Article</b> Aktionen oder bei <b>Launch</b> Aktionen. Macintosh Betrachter ignorieren den <b>File</b> Key, falls dieser Key gefunden wird.</p>
<b>DOSFile</b> 	string	<p>Bestimmt den MS-DOS Dateinamen im PDF-Format für Pfadnamen bei <b>GoToR</b> und <b>Article</b> Aktionen oder bei <b>Launch</b> Aktionen. Windows und DOS Betrachter ignorieren den <b>File</b> Key, falls dieser Key gefunden wird.</p>
<b>UnixFile</b> 	string	<p>Bestimmt den Unix Dateinamen im PDF-Format für Pfadnamen bei <b>GoToR</b> und <b>Article</b> Aktionen oder bei <b>Launch</b> Aktionen. Unix Betrachter ignorieren den <b>File</b> Key, falls dieser Key gefunden wird.</p> <p><i>Hinweis: Acrobat 2.0 Viewer suchen zuerst nach dem plattform-spezifischen Key, ist dieser nicht vorhanden wird der system-unabhängige Key File verwendet.</i></p>
<b>ID</b> 	array	<p>Ein Array aus zwei Strings, mit deren Hilfe eine PDF-Datei eindeutig identifiziert werden kann, wenn dieses über den Pfadnamen nicht möglich ist. Dieser Key ist optional. Auf Einzelheiten wird hier nicht eingegangen. (Nähere Informationen, vgl.: <i>Updates to the Portable Document Format Reference Manual</i>, Technical Note #5156, Adobe Systems.)</p>

- WinFile** <sup>2.0</sup> ▼ string Windows 2.0 Viewer verwenden die Windows Funktion ShellExecute um eine Anwendung zu öffnen. Die **WinFile**, **Dir**, **Op** und **Params** Keys korrespondieren mit den Parametern von ShellExecute. **WinFile** entspricht dem DOS-Dateinamen eines zu öffnenden Dokumentes oder einer zu öffnenden Anwendung.
- Op** <sup>2.0</sup> ▼ string Ein optionaler String, der die Operation unter **WinFile** definiert. Der Inhalt des Strings ist entweder „open“ oder „print“. Der Standard ist „open“. Ist der Wert des **WinFile** Key eine Anwendung und kein Dokument, so wird der **Op** Key ignoriert und die Anwendung geöffnet.
- Params** <sup>2.0</sup> ▼ string Mit diesem String können einer Windows Anwendung Parameter übergeben werden, wenn diese mit **Launch** geöffnet wird. Ist der Wert des **WinFile** Key ein Dokument, sollte der **Params** Key nicht verwendet werden.
- Dir** <sup>2.0</sup> ▼ string Optionaler String, der das Standardverzeichnis einer Windows Anwendung bestimmt, wenn diese mit **Launch** geöffnet wird.
- Unix** <sup>2.0</sup> ▼ string Definiert Unix Parameter für **Launch** Aktionen. Unix Viewer ignorieren **File** und **UnixFile** Keys, wenn dieser Key gefunden wird.

### 1.1.3 Namen und Parameter des View Arrays

Die folgende Tabelle definiert alle Namen und Parameter, die zusammen mit dem **View** Array verwendet werden können.

Name	Parameter	Bedeutung
<b>Fit</b>	Keine	Paßt die Seite dem Fenster an.
<b>FitB</b> 	Keine	Paßt den Inhalt der Bounding Box dem Fenster an. Die Bounding Box ist das kleinste Rechteck, das alle sichtbaren Elemente einer Seite enthält.
<b>FitH</b>	top	Paßt die Breite der Seite dem Fenster an. top gibt die Distanz vom Seitenursprung zur Oberkante des Fensters, in Point des User Space, an.
<b>FitBH</b> 	top	Paßt die Breite der Bounding Box dem Fenster an. top gibt die Distanz vom Seitenursprung zur Oberkante des Fensters, in Point des User Space, an.
<b>FitR</b>	$x_1y_1x_2y_2$	Paßt das durch die Parameter definierte Rechteck dem Fenster an. Haben die Entfernungen $x_1 - x_2$ und $y_1 - y_2$ unterschiedliche Vergrößerungsfaktoren, so wird der kleinere Faktor verwendet, um sicherzustellen, daß das Rechteck komplett dargestellt wird.
<b>FitV</b>	left	Paßt die Seitenhöhe dem Fenster an. left gibt die Distanz vom Seitenursprung zur linken Kante des Fensters, in Point des User Space, an.
<b>FitBV</b> 	left	Paßt die Höhe der Bounding Box dem Fenster an. left gibt die Distanz vom Seitenursprung zur linken Kante des Fensters, in Point des User Space, an.
<b>XYZ</b>	left top zoom	left und top geben die Entfernung des Seitenursprungs zur linken oberen Ecke des Fensters an. Ist left, top und zoom null, bleibt der aktuelle Wert dieses Parameters erhalten. Ein Ziel /View [/XYZ null null null] bewirkt einen Wechsel zur angebeben Seite und behält Position und Vergrößerungsfaktor der aktuellen Seite bei. Ein Vergrößerungsfaktor 0, hat die gleiche Bedeutung wie ein Vergrößerungsfaktor null.

*Hinweise: – Angaben für `left` und `top` sind nur dann sinnvoll, wenn gilt: `left`, `(top) > x`, `(y)`, und zwar bei bereits erfüllter Fit-Vergrößerung und mindestens ausgefülltem Fenster, wobei `x`, `y` sichtbare Koordinaten im Fenster sind. (Vgl. Beispiel-Dokument `View.pdf` auf Diskette.)*

*– Der Name **FitW** (Anpassung der Breite einer Seite an das Fenster), welcher teilweise in der Literatur aufgeführt ist, wird von Acrobat 2.0 Viewern nicht unterstützt.*

## 1.2 Programmbeispiele

Das folgende Kapitel enthält einige Beispiele, um zu zeigen, wie der **pdfmark** Operator verwendet werden kann.

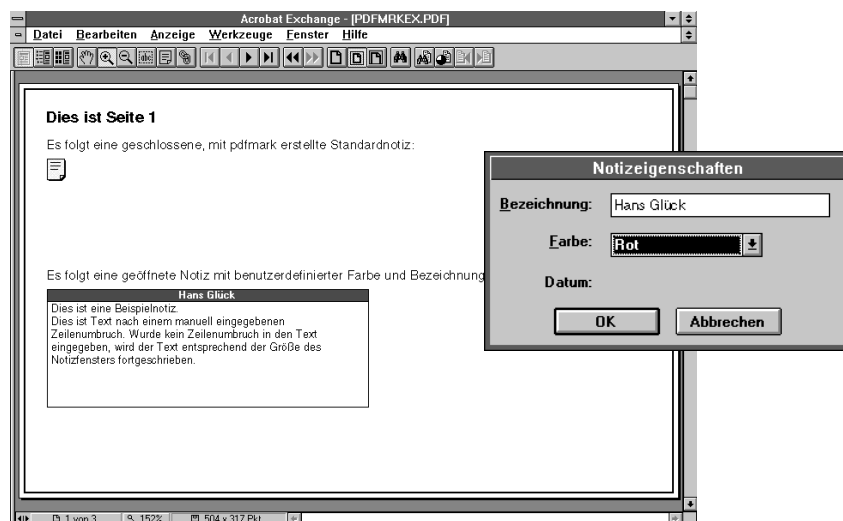
### 1.2.1 Notizen

Notizen benutzen den PDFDocEncoding-Vektor. Werden Strings über mehrere Zeilen fortgeschrieben (siehe Beispiel), so sind die Zeilen mit `\` abzuschließen. Der Backslash bewirkt, daß ein Carriage Return am Zeilenende sowie der Backslash selbst ignoriert werden. Ein Zeilenvorschub innerhalb eines Strings ist mit `\r` zu erreichen.

```
% NOTIZ
[ /Rect [ 75 425 450 563 ]
  /Open true
  /Title (Hans Gl\374ck)
  /Contents (Dies ist eine Beispielnotiz. \r
Dies ist Text nach einem manuell \
einggegebenen Zeilenumbruch. Wurde kein \
Zeilenumbruch in den Text eingegeben, \
wird der Text entsprechend der Gr\366\337e \
des Notizfensters fortgeschrieben.)
  /Color [1 0 0]
  /Border [0 0 1]
/ANN pdfmark
```

### 1.2.2 Verknüpfungen

Abb. 1.1  
Notiz und  
Notizeigenschaften



Verknüpfungen können verschiedene Teile eines Dokumentes miteinander verbinden. Sinnvoll ist dies z.B. bei Inhaltsverzeichnissen, Indizes, Verweisen oder Grafiken. Ein Beispiel für eine Verknüpfung:

```
% VERKNÜPFUNG
[   /Rect [ 70 550 245 575 ]
    /Border [ 0 0 2 [ 3 ] ]
    /Color [0 1 0]
    /Page /Next
    /View [ /XYZ -5 797 1.5]
  /LNK pdfmark
```

Die Verknüpfung in einem Dokument:

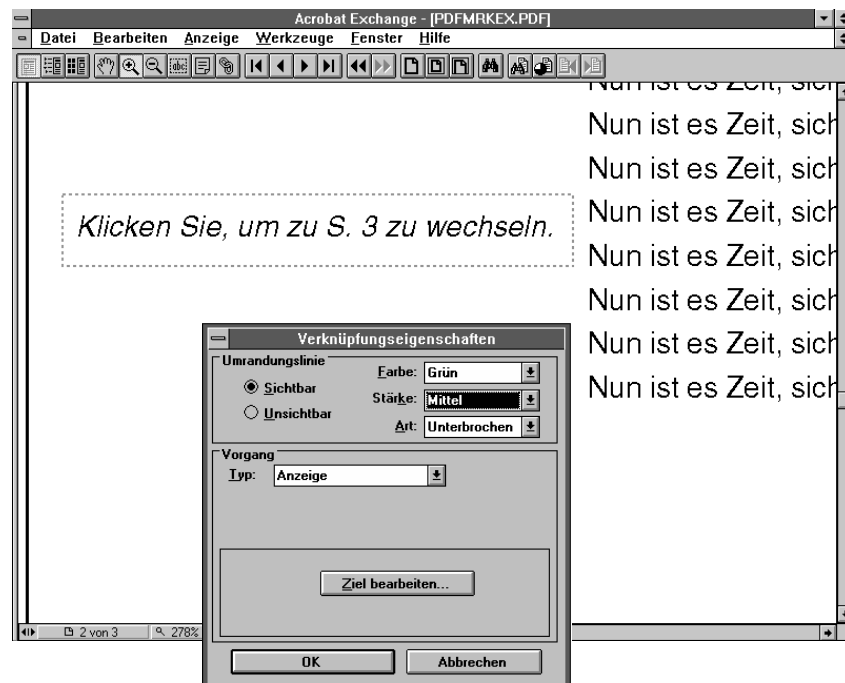


Abb. 1.2  
Verknüpfung und  
Eigenschaften der  
Verknüpfung

### 1.2.3 Lesezeichen

Lesezeichen sind identisch mit Verknüpfungen. Sie befinden sich jedoch außerhalb des Dokumentes, in einem separaten Fenster. Hilfreich sind Lesezeichen, wenn kein Inhaltsverzeichnis im Dokument selbst vorhanden ist. Es folgt ein Beispiel für ein Lesezeichen, welches drei untergeordnete Lesezeichen enthält:

```

[/Count 3
 /Page 1
 /View [/XYZ 44 730 1.0]
 /Title (Inhaltsverzeichnis Nr. 1)
 /OUT pdfmark

[/Page 1
 /View [/XYZ 44 730 1.0]
 /Title (Seite 1 - 100%)
 /OUT pdfmark

[/Page 2
 /View [/XYZ 44 730 2.25]
 /Title (Seite 2 - 225%)
 /OUT pdfmark

[/Page 3
 /View [/Fit]
 /Title (Seite 3 - Fenstergr\366\337e)
 /OUT pdfmark
    
```

Die geöffneten Lesezeichen in einem Dokument:

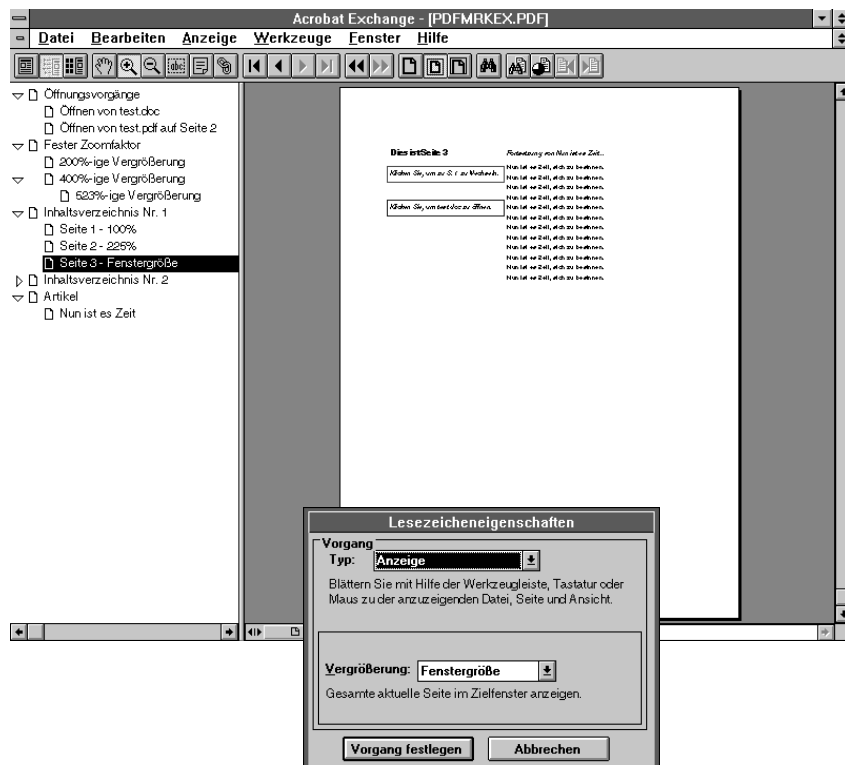


Abb. 1.3  
Lesezeichen mit  
Eigenschaften des  
Lesezeichens „Seite 3  
Fenstergröße“

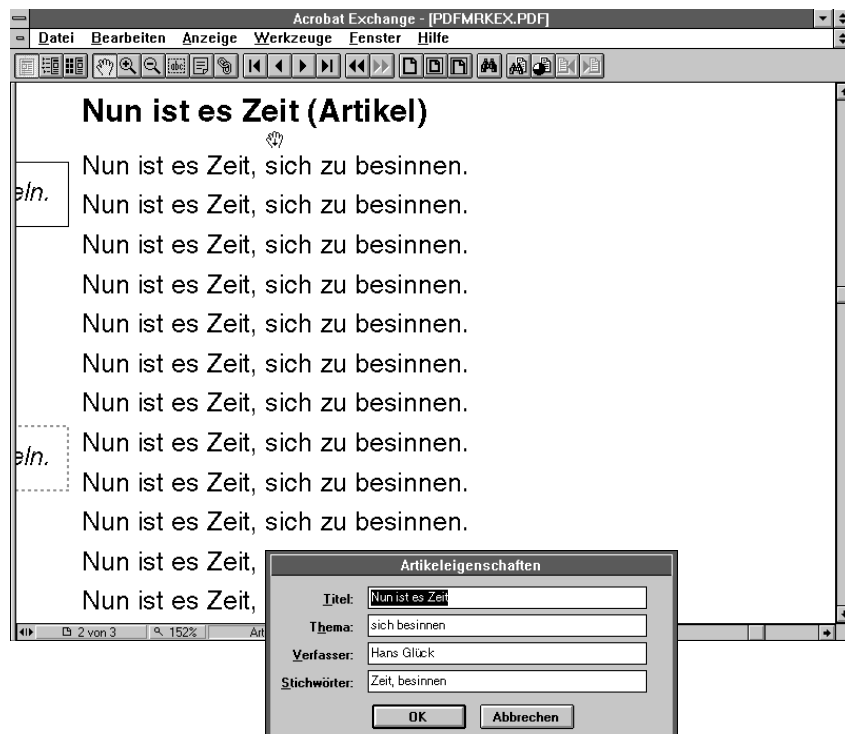
## 1.2.4 Artikel

Artikel erleichtern das Lesen in Dokumenten, die nicht speziell für den Bildschirm erstellt wurden. Der Leser wird mit Hilfe der Artikel durch das PDF-Dokument geführt, ohne ständig scrollen oder vergrößern zu müssen. So können z.B. die Spalten einer Drucksache als zusammenhängender Textfluß definiert werden. Ein Beispiel eines Artikels, der sich über zwei Seiten fortsetzt:

```
% ARTIKELABSCHNITT
[ /Title (Nun ist es Zeit)
  /Author (Hans Gl\374ck)
  /Subject (sich besinnen)
  /Keywords (Zeit, besinnen)
  /Rect [ 225 500 535 705 ]
  /Page 2
/ARTICLE pdfmark

[ /Title (Nun ist es Zeit)
  /Rect [ 225 500 535 705 ]
  /Page 3
/ARTICLE pdfmark
```

Abb. 1.4  
Artikel und  
Artikeleigenschaften



### 1.2.5 Seitenbeschnitt

Wurde ein Dokument nicht für den Bildschirm, sondern zur Ausgabe auf einen Belichter erstellt, so enthält es meist Beschnittmarken. Die Seiten müssen daher nachträglich beschnitten werden. Es folgt ein Beispiel eines manuell programmierten Seitenbeschnitts:

```
% ALLE SEITEN BESCHNEIDEN
[ /CropBox [54 403 558 720]
/PAGES pdfmark
```

Das Ergebnis dieses Seitenbeschnitts in einem PDF-Dokument:

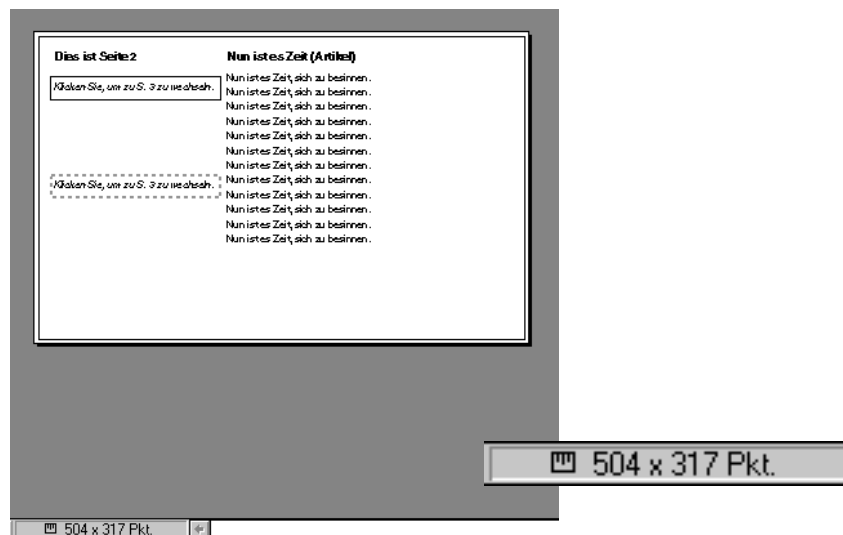


Abb. 1.5  
Seitenbeschnitt

### 1.2.6 Info und Catalog

Mit Hilfe der Namen **DOCVIEW** und **DOCINFO** können Informationen in das Info und Catalog Dictionary eingetragen werden. Das Info Dictionary enthält z.B. folgende Key-Value Paare:

```
% CATALOG-INFO
[ /Title (Mein Testdokument)
  /Author (Hans Gl\374ck)
  /Subject (pdfmark 2.0)
  /Keywords (pdfmark, Beispiel, Test)
  /Creator (manuell programmiert)
  /ModDate (D:19940912205731)
  /Mykey (Meine privaten Infos)
/DOCINFO pdfmark
```

Diese Informationen sind in Acrobat Viewern im Menü *Datei* unter *Dokumenteninfo*, *Allgemein* abrufbar:



Abb. 1.6  
Info Dictionary

Das Catalog Dictionary enthält Informationen zum Öffnen eines Dokumentes:

```
% INFORMATIONEN ZUM ÖFFNEN DER DATEI
[   /PageMode /UseOutlines
    /Page 2
    /View [ /XYZ null null null ]
/DOCVIEW pdfmark
```

Diese Informationen sind im Menü *Datei* unter *Dokumenteninfo*, *Öffnen* abrufbar:

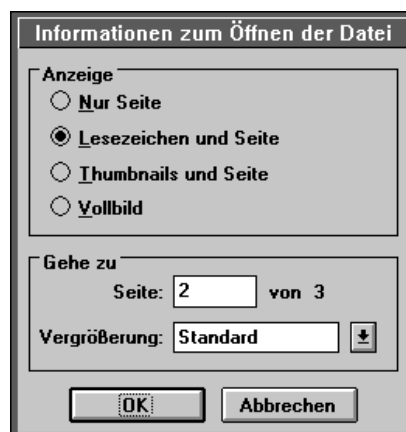


Abb. 1.7  
Catalog Dictionary