

[1] pdfmark operatorとは?

pdfmark operatorを利用することで、Acrobat DistillerがPostScriptファイルをDistillして、PDFファイルを作成する過程で、あらかじめPostScriptファイルにある特別なコマンド(pdfmark operator)を追加し、Acrobat Exchangeで行うしおり、サムネール、ノート、アーティクル、書類情報、ムービーファイルの貼り付け、ページクロッピング、初期設定等のあらゆる設定をディスティル時に組み込み処理させようとするものです。ここでは、この特別な記述(pdfmark operator)について解説するとともに、実際の使用方法を詳しく解説してゆきます。

pdfmark operatorとは、一種のプログラミング言語であり、PostScriptファイル中にこれらのpdfmark operatorを埋め込みます。埋め込まれたそれぞれのpdfmark operatorは単にマーカーとして参照されるだけで、実際のプリンターが搭載しているPostScriptインタプリタではスキップされるように記述するため、印刷には影響がありません。

[2] pdfmark operatorの種類と概略

pdfmark operatorはmark object、variable number of key-value pairs、及び、name objectをその主題として得ますが、値は返しません。PostScriptファイル中のpdfmark operatorのそれぞれの手続きは単にマーカーとして参照されます。

一般的なpdfmark operatorの文法は以下のようになります。

[... Various key-value pairs ... KIND pdfmark	"["でpdfmark operatorの記述が始まり、"... Various key-value pairs ..."の部分に各々のpdfmark operatorが記述されます。"pdfmark"で一つのoperatorの終了です。
--	---

"KIND"はpdfmarkの種類を特定する名前です、主に以下の種類があります。

/ANN	リンク、ノート、ムービー、サウンドファイル、等のオブジェクトを貼り付ける場合に使用されます。
/OUT	しおりを作成する場合に使用します。
/ARTICLE	アーティクルを作成する場合に使用します。
/DEST	Webページや他のPDFファイルから特定のページへのリンクアンカーを設定する場合に使用します。
/PS	ポストスクリプトファイルの制御コマンドを記述する場合に使用します。
/DOCINFO	作成者・書類タイトル・作成日等の書類情報を作成する場合に使用します。
/DOCVIEW	書類を開いたときのページモード(しおり+ページ、サムネール+ページ、フルスクリーン表示、等)や最初に表示するページを規定する場合に使用します。
/PAGE(S)	ページをクロッピング(余白の切り取り)する場合に使用します。
/PUT	書類Dictionary、ページDictionaryに値を書き込む場合に使用します。ページトランジションの指定や、特定ページに貼り込まれたムービーを自動再生させる場合などで、そのページのDictionaryに命令を書き込みます。

[3] pdfmarkの構成と配置方法

プリンタ等のPostScript機器が上記のような記述を含むファイルを実行しないことを確実にするため、以下のようなPostScript languageコードをPostScriptファイル中のプロローグ部に配置しなければなりません。これは、PostScriptインタプリタがファイルを処理するとき、pdfmark operatorを実行しないようにするもので、以下のように指定されています。

```
/pdfmark where
{POP} {userdict /pdfmark /cleartomark load put} ifelse
```

作成したpdfmarkを配置する方法には2通りあります。一つはPDFファイルにする書類で書き出されたPostScriptファイルに直接書き込んで、DistillerでPDF変換する方法と、もう一つはその書類中にEPSファイルとして割り付けた上で、PostScriptファイルに書き出し、それをPDFファイルに変換する方法です。この場合には書類を作成するアプリケーションソフトがEPSファイルの割付に対応していることが前提となります。

[3-1] PostScriptファイル中に配置する場合の記述

pdfmark operator使用宣言

以下のコードをPostScriptファイルのプロローグ部に挿入し、これからpdfmark operatorを使用しますよと宣言します。これは、前述のようにプリンター等のPostScriptインタプリタがファイル中に配置されたpdfmark operatorコマンドを無視するように定義するものでもあります。プロローグ部と言っても何処から何処までかは専門家でないとは分かりませんが、取りあえず文頭に挿入しておけば間違いがないでしょう。

```
%!PS-Adobe-3.0
%%BeginPrologue
    /bd {bind def} bind def
    /fsd {findfont exch scalefont def} bd
    /sms {setfont moveto show} bd
    /ms {moveto show} bd
    /pdfmark where
        {pop} {userdict /pdfmark /cleartomark load put} ifelse
%%EndProlog
%%Begin Setup
```

この宣言文に続けて、各々のpdfmark operatorを配置します。

最後に、PostScript言語のファイル終了コマンド[%%EOF]を埋め込みます。

具体的な一例としては以下のような記述となります。リンクアノテーションを2ページに貼り付け、リンク先を3ページ目に設定するpdfmarkです。

```
%----- pdfmark使用宣言部 PS-----
%!PS-Adobe-3.0                                     %-----PSファイルヘッダ部
%%BeginPrologue
    /bd {bind def} bind def
    /fsd {findfont exch scalefont def} bd
    /sms {setfont moveto show} bd
    /ms {moveto show} bd
    /pdfmark where
    {pop} {userdict /pdfmark /cleartomark load put} ifelse
```

```

%%EndProlog
%%Begin Setup

%----- リンクアノテーション pdfmark operator の記述 -----
[
    /Rect [ 100 100 300 200 ]
    /Border [ 0 0 1]
    /Color [0 1 0]
    /SrcPg 2
    /Page 3
    /Subtype
    /Link
    /ANN
pdfmark
%-----pdfmark の終了-----
%%EOF

```

PostScriptファイルに書き出された書類をテキストエディタで開いて、このマーカーをファイルの先頭部に挿入して、Distillerで開いてPDFファイルを作成します。すると、2ページ目にリンクオブジェクトが配置されるでしょう。

[3-2] EPSファイルとして作成し、書類中に割り付けて使用する

作成したpdfmark operatorを記述したテキストファイルを拡張子".eps"にして保存します。DTPアプリケーション(QuarkXPress, PageMaker)、イラストレータなどのEPSファイルに対応したソフトで作成した書類にこのpdfmarkを貼り付けます。

EPSファイルの記述はPostScriptファイルの場合とヘッダ部のみが異なり、以下の通りとなります。

```

%----- pdfmark使用宣言部 EPS-----
%!PS-Adobe-3.0 EPSF-3.0                               %-----EPSファイルヘッダ部
%%BoundingBox: 0 0 9 9
systemdict /pdfmark known not {userdict /pdfmark systemdict /cleartomark get put } if

%----- リンクアノテーション pdfmark operator の記述 -----
[
    /Rect [ 100 100 300 200 ]
    /Border [ 0 0 1]
    /Color [0 1 0]
    /SrcPg 2
    /Page 3
    /Subtype
    /Link
    /ANN
pdfmark
%-----pdfmark の終了-----
%%EOF

```

これを記述したテキストファイルを適当な名前+.epsで保存し、書類中の任意の位置に割付けます。割り付けた書類上では、上記ヘッダ部で指定された(BoundingBox: 0 0 9 9) 矩形図形が見えるでしょう(この場合では9ポイント角の矩形ができます)。しかし、上記PostScriptファイルの場合と同じように印刷時にはこれらの記述は無視されるため、印刷はされません。

注意)

/SrcPg キーを使用して貼り付けるページを指定した場合は、貼り付ける位置(ページ)に関係なくここで規定したページにリンクが作成されます。従って、EPSファイルを割り付けたページにリンクを埋め込むた

めには、このマーカーを削除しておく必要があります。逆に言うと、どのページにこのEPSファイルが割り付けられても、貼付けされるリンクのページはここで記述されることとなりますので、どこに割り付けても良いこととなります。

以上がPostScriptファイル形式、及び、EPSファイル形式でのpdfmarkの記述方法です。次に各々のマーカーについて具体的に解説してゆきます。

[4] 各々のpdfmarkの記述

[4-1] ノート

Annotationマーカーはノート、リンク及びカスタムアノテーションを作成します。アノテーションは"ANN"の名前でpdfmark operatorを使用することが指定されています。

ノート作成の記述は：

```
[
  /Contents String
  /Rect [llx lly urx ury]
  /SrcPg pagenum
  /Open boolean
  /Color array
  /Title String
  /ModData datestring
  /Subtype string
/ANN pdfmark
```

Key	Type	意味
Contents	string	(必須) ノート中に記載する文字列。最大65,535文字。
Rect	array	(必須) [Xll Yll Xur Yur]の4つの数値で記述。[Xll Yll Xur Yur]とは四角形で表示されるノートの左下(x, y)と右上(x, y)を表す。lower-left x, y upper-right x, y
SrcPg	Integer	(オプション) ノートを表しするページ番号。書類の最初のページがページ1となる。このSrcPgはPostScriptファイル中のどこに記載しても良い。もしSrcPgを使用しなかった場合は、PostScriptファイル中で、このソースが記載しているところに該当するページにノートが表示されます。
Open	boolean	(オプション) trueでは、ファイルをあけたとき、ノートも開けられて表示されます。falseでは、ノートはアイコンとして閉じられています。もしOpenコマンドを使用しない場合は、falseの場合と同じであるが、オープンアクセス制限を持ちません。
Color	array	(オプション) ノートアイコンの背景色、開いたノートウィンドウのタイトルバーの色、及び、フレームの色を指定します。色の指定は、DeviceRGB color spaceで規定される、0から1までの数字で表されます。このコマンドがない場合は初期設定の色が適用されません。
Title	string	(オプション) ノートのタイトル。ノートの最大長は65,535文字ですが、通常は32文字以内にします。
ModDate	string	(オプション) ノート作成時の日時を記載します。以下のようなフォームでなければなりません (D:YYYYMMDDHHmmSSOHH'mm') "D:"はオプション前置詞、YYYYは西暦、MMは月(01-12)、DDは日(01-31)、HHは時間(00-23)、mmは分(00-59)、SSは秒(00-59)を表します。文字列の残りは、GMTに対するローカルタイムの関係を定義します。0はGMTに対し進んでいる場合は"+", 遅れている場合は"- "を表します。HH'はGMTの時間から絶対的なズレを表し、mm'分のズレを表します。もし、GMTが規定されていない場合は、指定された時間とGMTの関係が不明となります。GMTが規定されているかどうかに関係なく、文字列の残りはローカルタイムを規定すべきです。
Subtype	name	(オプション) アノテーションのPDFサブタイプ。使用する場合は/Textで記載されます。

サンプルコード

```
[ /Rect [75 425 350 563]
  /Open true
  /Title (Sample Note #1)
  /Contents (This is a sample of note.  ¥nHere is some text after a forced line break.)
  /Color [1 0 0]
  /SrcPg 2
  /ANN
```

pdfmark

解説

座標(75, 425, 350, 563)のウインドウを、開いた状態で表示し、ノートの内容は(This is a line break.)¥nはノート中で強制改行をするコマンドです。ノートアイコン及び、ノートウインドウのタイトルバーの色は100%の赤です。このノートはPDFファイルの2ページ目に貼り付けられます。こんな感じかな？

Sample Note #1

This is a sample of note.

Here is some text after a forced line break.

カラーを変えたい場合は、/Color [R G B]に対応して[0.7 0 1]などと指定します。ノートアイコンを初期設定で開きたくない場合は、/Open falseとします。

(Tips)

/Rectで指定される座標

/Rectで指定される座標は[lly(low-left y), llx(low-left x), ury(upper-right y), urx(upper-right x)]で指定され、原点はページの左下である。単位はポイントで表され、1ポイントは72/1インチである。A4の用紙の場合では、横:595ポイント(210mm)、縦:841ポイント(297mm)になる。A4用紙いっぱいの4角形を指定したい場合は[0 0 595 841]になります。

因みに、PDFファイルで最大取り扱えるサイズは、45インチX45インチ(3240 x 3240ポイント)です。最小サイズは1インチX1インチ(72 x 72ポイント)です。

[4-2]リンク

リンク作成の記述は：

```
[
  /Rect    [llx llr urx ury]
  /Border  [bx by x [d]]
  /SrcPg  pagenum
  /Color  array
  /Subtype /Link
  ..Action-specifying key-value pairs...
/ANN pdfmark
```

Key	Type	意味
Rect	array	(必須) [Xll Yll Xur Yur]の4つの数値で記述。[Xll Yll Xur Yur]とは四角形で表示されるリンクボタン枠の左下(x, y)と右上(x, y)を表す。lower-left x, y upper-right x, y
Border	array	(オプション) リンク境界線の属性。Borderは3つの数字とオプションでなる。Borderが[bx, by, c]であるとする、(bx)は水平コーナーの半径、(by)は垂直コーナーの半径、(c)は境界線の実線の太さを表します。Borderが[bx, by, c[d]]示される場合は、境界線が波線で、4番目の要素(d)は波線の長さと同隔を表します。初期設定では/Border [0 0 1]です。つまり、1ポイント境界線を持つ4角形となります。
SrcPg	Integer	(オプション) リンク先のページ番号。書類の最初のページがページ1となる。このSrcPgがあるとリンクマーカはPostScriptファイル中のどこに記載しても良い。このキーを含まないリンクマーカはリンクソースページのPostScriptファイルの範囲内になければならない。
Subtype	name	(必須) アノテーションのPDFサブタイプ。/Linkでなければなりません。

サンプルコード

```
% Fancy link
[ /Rect [ 70 550 210 575 ]
  /Border [ 0 0 2 [ 3 ] ]
  /Color [0 1 0]
  /Page /Next
  /View [ /XYZ -5 797 1.5 ]
  /Subtype /Link
/ANN
pdfmark

% Link
[ /Rect [ 70 650 210 675 ]
  /Border [ 16 16 1 ]
  /Color [1 0 0]
  /SrcPg 2
  /Page 1
  /View [ /FitH 5]
  /Subtype /Link
/ANN
pdfmark

% Link to a named destination
[ /Rect [ 70 650 210 675 ]
  /Border [ 16 16 1 [ 3 10 ] ]
```

```
/Color [ 0 .7 1 ]
/SrcPg 2
/Dest /MyNamedDest
/Subtype /Link
/ANN
```

pdfmark

```
% Link that launches another file
```

```
[ /Rect [ 70 600 210 625 ]
  /Border [ 16 16 1 ]
  /SrcPg 2
  /Color [ 0 0 1 ]
  /Action /Launch
  /File (test.doc)
  /Subtype /Link
  /ANN
```

pdfmark

```
% Custom Link action (URI link for the Acrobat WebLink plug-in)
```

```
[ /Rect [ 50 425 295 445 ]
  /Action << /Subtype /URI /URI (http://www.adobe.com) >>
  /Border [ 0 0 2 ]
  /SrcPg 2
  /Color [ .7 0 0 ]
  /Subtype /Link
  /ANN
```

pdfmark

解説

```
[ /Rect [ 70 550 210 575 ]
  /Border [ 0 0 2 [ 3 ] ]
  /Color [ 0 1 0 ]
```

座標 (70(llx) 550(lly) 210(urx) 575(ury))の範囲で、緑色の境界線太さ2ポイントの間隔3の破線の矩形のリンクを作成する。

```
/SrcPg 2
```

2ページ目にリンクを貼り付ける。

```
/Page /Next
/View [ /XYZ -5 797 1.5 ]
```

リンク先はソースコードが配置されている次のページへ。150%の拡大表示する。

```
/Dest /MyNamedDest
```

ファイル中にアンカーされた"/MyNamedDest"へのリンク。このアンカーネームが設定されていないとDistill時にWarningされるでしょう。PDF中のアンカーネームは後述します。

```
/Action /Launch
/File (test.doc)
```

test.docの名前の外部書類を起動する。

```
/Action << /Subtype /URI /URI (http://www.adobe.com) >>
```

webブラウザを起動させ指定したURL(http://www.adobe.com)へジャンプ。

(チップス!)

これは便利!

このLink Annotationを利用して最も効率が上がるPDF作成の一例を紹介します。上記コードでは紹介しませんでした。リンクを貼り付けるページをコード上で記述することもできます。例えば、数十ページ、数百ページにわたるPDFファイルで各ページの同じ場所にリンクを作成し、リンク先がそのPDFファイル中の特定のページ(例えば目次とか、章のはじめとか)または、特定のURLへリンクを作成するような場合では、大変な威力を発揮するでしょう。以下のようなコードをそのページ分作成すればいいのです。

```
[ /Rect [ 70 550 210 575 ] /Border [ 0 0 1 ] /Color [ 0 1 0 ] /SrcPg 1 /Page 1 /Subtype /Link /ANN pdfmark
[ /Rect [ 70 550 210 575 ] /Border [ 0 0 1 ] /Color [ 0 1 0 ] /SrcPg 2 /Page 1 /Subtype /Link /ANN pdfmark
[ /Rect [ 70 550 210 575 ] /Border [ 0 0 1 ] /Color [ 0 1 0 ] /SrcPg 3 /Page 1 /Subtype /Link /ANN pdfmark
[ /Rect [ 70 550 210 575 ] /Border [ 0 0 1 ] /Color [ 0 1 0 ] /SrcPg 4 /Page 1 /Subtype /Link /ANN pdfmark
```

/SrcPg がリンクを張るソースページになります。ここでは"1"としています。/Pageがリンク先のページ番号になります。/SrcPgのソース番号を変えただけの上記コードを続けて作成します。これらのコードをPostScriptファイルのプロローグ部で挿入したpdfmark宣言文に続けて記述するといいいでしょう。この方法であれば、PostScriptファイルのページ範囲を調べる必要もありません。

記述します。これらのアノテーションはリンクのように書類中に埋め込まれて表示されます。アクティベーションエリアは不表示で、リンクボタンと同様の境界線、そして、ムービーのポスターフレーム表示を持ちます。ムービーを表示・再生させるためのいくつかの方法を制御するオプションがあります。

ムービープレーヤーアノテーション属性

Key	Type	意味
Subtype	name	(必須) アノテーションサブタイプ 常にMovie
Movie	dictionary	(必須) ムービーの静的特性の記述、以下テーブルを参照
A (Activiation)	Boolean 又は	(オプション) アノテーション矩形範囲内をクリックすることでムービーを見せるか見せないかの表示のフラグ。値は： false クリックしてもムービーは再生しない true 初期アクティベーション値でムービーの再生
	Dictionary	(オプション) ムービーの再生指示、下記テーブル参照

ムービーディクショナリはムービーデータを配置するために必要な情報とアノテーション矩形範囲にポスター（もし、リクエストするなら）を表示するための情報を持ちます。

ムービーアノテーション属性

Key	Type	意味
F (File)	File specification	(必須) ムービーファイルファイル名&相対パス指定 外部ファイルを指定する場合には、PDFファイル本体からの相対パスで指定されなければならない。
Aspect	array	(オプション) もしムービーがvisibleの場合、ムービーの矩形範囲の水平・垂直サイズ 単位はピクセル [horiz vert]。 "invisible movie"とはサウンドのみを持つ
Rotate	integer	(オプション) ムービーを回転させなければならない角度を時計回りで規定、これは、ページが回転された場合に関係します。この値は90度単位で記述し、初期設定は0度になっています。
Poster	Boolean または	(オプション) ポスターを表示するためにムービーファイルから取り出すか否かを表示するフラグ。 false ポスターイメージを表示しない（これは初期設定です） true ムービーファイルからのポスターイメージを表示する
	stream	(オプション) ポスターとして表示するためのイメージリソース

ムービーアクティベーションディクショナリーはムービー再生の動的制御を必要とする情報を持ちます。

ムービーアクティベーション属性

Key	Type	意味
ShowCont		

Start	string, or array	最初からスタートする。 ムービー時間の値は64ビット整数が基本となる。
Duration	number, string, or array	(オプション)ムービーセグメント再生の期間、無い場合は、最後まで再生される。 ムービー時間の値は64ビット整数が基本となる
Rate	number	(オプション)ムービー再生の初期スピード、初期設定は1.0。
Volume	number	(オプション)ムービーの初期音量設定、このnumberは-1から1の間でなければならない。負の設定はミュートされる。初期設定は1です。
FWScale	array	(オプション)フローティング再生ウィンドウのスケール。FWScaleは2個の整数のarrayで記述：[numerator denominator] の表示はムービーの拡大比率を表わす。ムービーの最終のウィンドウサイズは (numerator ÷ demoniator) × Aspect ピクセルとなる。FWScaleキーがあることはムービーがフローティングウィンドウで再生されることを意味します。 FWScaleキーが無いことはムービーがアノテーション矩形範囲で再生されることを意味します。
FWPosition	array	(オプション)フローティングウィンドウ用としてのムービーの画面上の位置、FWPositionは2個の整数のarrayで記述：[horizontal vertical]の表示は画面の左部、上部への関係でムービーウィンドウの位置を表示する。それぞれのnumberは0から1の範囲でなければならない。[0.5 0.5]の記述は画面の中央でムービーウィンドウを表示することを意味します。

サウンドアノテーション

サウンドアノテーションはサウンドファイルからの入力やコンピュータのマイクからの録音のような音声情報を含むだけでテキストアノテーションと同じです。アノテーションは通常テキストアノテーションのようで、サウンドアノテーションであることを表示するアイコン(スピーカー)のみが異なっています。

サウンドアノテーション属性

Key	Type	意味
Subtype	name	(必須)アノテーションサブタイプ、常にSound
Sound	stream or File specification	(必須)サウンドオブジェクト(stream)又は、外部サウンドファイル(file specification)

```

サンプルコード

%-- 3ページにムービー貼り付け
[
  /Rect [150 150 450 350]
  /SrcPg 3
  /Subtype /Movie
  /Title (kt)
  /Movie << /F (folder/kt.mov) /Poster true>>
  /A << /Showcontrols false /Mode /Open /Volume 0.8 >>
  /F 1
  /Border [0 0 10[5]]
  /Color [0 1 0.5]
  /ANN pdfmark

%-- 1ページにムービー貼り付け
[
  /Rect [150 100 450 280]
  /SrcPg 1
  /Subtype /Movie
  /Title (anibird)
  /Movie << /F (bird.mov) /Poster false >>
  /F 1

```

(チップス!)

key-value pair で何だ

今までに説明してきましたpdfmark operatorの記述で、key-value pairと言う記述がありますが、これはどういう意味で使用されているだろうか？

これはキーに対する値のセットを記述するということで、例えば、/SrcPgのキーに対する値を1とか、/Subtypeキーの対しては/Movieの値、/Posterキーではfalseの値、/Volumeキーでは0.8の値を記述することです。

ついでながら、上記テーブル中のKeyに対するTypeの記述では、boolean、number、integer、string、name、array、または、dictionary等が記述されていますが、具体的には以下のような記述になります。

boolean	(真か偽) true or false
number	(個数) 1、2、3
integer	(整数) 1、2、3
string	(文字列) This is a sample pdfmark code. など
name	(名称) Movie、Sound、UseOutlinesなど
array	(複合) [85 120 450 550]、[110 20 1.5]、[/XYZ 1 1 3]など
dictionary	(辞書) << /Subtype /URI /URI (http://www.adobe.com) >> 、/Movie << /F (bird.mov) /Poster false >> など

[4-4] しおり

PDFのアウトライン入力として知られるBookmark(しおり)は、"OUT"の名前でpdfmark operatorを使用することが指定されています。

しおり作成の記述は：

```
[ /Title String
  /Count int
  ...Action-specifying key-value pairs ...
  /OUT
pdfmark
```

Key	Type	意味
Title	string	(必須) しおりの文字。Titleは最大65,535文字入力可能。標準は32文字以内が推奨されま す。キャラクタセットとエンコーディング方式はPDFDocEncodingを使用。
Count	interger	(階層を持ったしおりを作成する場合には必須) 以下のルールに従い、階層を持つしおりの 表示と番号を規定している。 1. しおりのマーカーはPostScriptファイル中の順番で現れなければならない。 2. 階層を持たないしおりは"Count"キーを持たない。 3. 階層を持つしおりは"Count"キーを持たなければならない。このキーの絶対の数値 は、階層第一レベルのしおりの数となる。値が正の場合は、階層下のしおりは開かれ る。負の場合は、閉じられる。ここで、開かれたしおりとは、その階層を表示する事 で、閉ざされたしおりはその階層を表示しないことである。

さらに、Key value pairsはアクションコマンドを持ちます。つまり、しおりに何らかのアクション(他の
ファイルを読み込む、アプリケーションを起動する、又は、指定するURLへジャンプする等)を設定すること
が可能です。これについては後ほど説明します。

このBookmark markerは"Count"キーによる構成順序を除いては、PostScriptファイル中のどこに記載しても
よいことになっています。

```
サンプルコード

[/Count 2 /Page 1 /View [/XYZ 44 730 1.0] /Title (Open Actions) /OUT pdfmark
  [/Action /Launch /File (test.doc) /Title (Open test.doc) /OUT pdfmark
  [/Action /GoToR /File (test.pdf) /Page 2 /View [/FitR 30 648 209 761]
  /Title (Open test.pdf on page 2) /OUT pdfmark

[/Count 2 /Page 2 /View [/XYZ 44 730 1.0] /Title (Fixed Zoom) /OUT pdfmark
  [/Page 2 /View [/XYZ 44 730 2.0] /Title (200% Magnification) /OUT pdfmark
  [/Count 1 /Page 2 /View [/XYZ 44 730 4.0] /Title (400% Magnification) /OUT pdfmark
  [/Page 2 /View [/XYZ 44 730 5.23] /Title (523% Magnification) /OUT pdfmark

[/Count 3 /Page 1 /View [/XYZ 44 730 1.0] /Title (Table of Contents #1) /OUT pdfmark
  [/Page 1 /View [/XYZ 44 730 1.0] /Title (Page 1 - 100%) /OUT pdfmark
  [/Page 2 /View [/XYZ 44 730 2.25] /Title (Page 2 - 225%) /OUT pdfmark
  [/Page 3 /View [/Fit] /Title (Page 3 - Fit Page) /OUT pdfmark

[/Count -3 /Page 1 /View [/XYZ 44 730 1.0] /Title (Table of Contents #2) /OUT pdfmark
  [/Page 1 /View [/XYZ null null 0] /Title (Page 1 - Inherit) /OUT pdfmark
  [/Page 2 /View [/XYZ null null 0] /Title (Page 2 - Inherit) /OUT pdfmark
```

%Bookmark with a URI as an action

```
[/Count 0 /Title (Adobe's Home page)
/Action << /Subtype /URI /URI (http://www.adobe.com)>> /OUT pdfmark
```

解説

[/Count 2 /Page 1 /View [/XYZ 44 730 1.0] /Title (Open Actions) /OUT pdfmark
このしおりは、"Open Actions"というタイトルで、2つの下位しおりを持ち、1ページ目にリンクされる。表示モード/View [/XYZ 44 730 1.0]は、44(top) 730(left)の場所で、100%の倍率で表示する。この記述は[アクションの項目](#)で解説します。

[/Action /Launch /File (test.doc) /Title (Open test.doc) /OUT pdfmark
"Open test.doc"のしおりを作成し、外部ファイル(test.doc)の書類を外部アプリケーションで表示させる。

[/Action /GoToR /File (test.pdf) /Page 2 /View [/FitR 30 648 209 761] /Title (Open test.pdf on page 2) /OUT pdfmark
"Open test.pdf on page 2"のしおりを作成し、リンク先は外部ファイル(test.pdf)の2ページ目である。表示モードは、30(x1) 648(y1) 209(x2) 761(y2)の座標で指定される。/View [/FitR 30 648 209 761]の詳細についても[アクションの項目](#)で解説します。

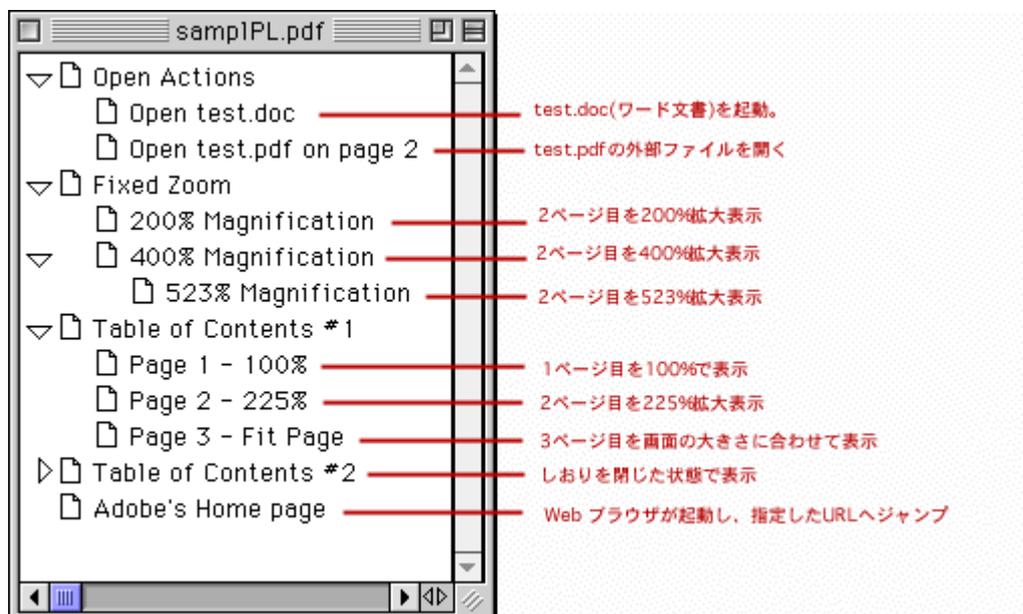
中略

....
....

[/Count -3 /Page 1 /View [/XYZ 44 730 1.0] /Title (Table of Contents #2) /OUT pdfmark
/Count -3は3つの下位しおりを持ち、それをデフォルトで表示しない。

[/Count 0 /Title (Adobe's Home page)
/Action <</Subtype /URI /URI (http://www.adobe.com)>> /OUT pdfmark
これはしおりにアクションとしてURLを埋め込んだ例です。

このファイルをPostScriptファイル中に追加し、Distillを行うと、以下のようなしおりが作成されるでしょう。



ファイルを開けたときの表示方法(しおりを表示する、サムネールを表示する、またはどちらも表示しない)

については「表示モード」の項目を参照ください。
日本語の使用はこのまま書き込んでもエンコード方式が異なるためDistillerが理解できませんので注意ください。しおり、ノート、書類情報などに日本語を使用するためにはPDFDocEncodingに沿ってエスケープ処理された文字列を使用する必要があります。詳細は「[pdfmarkで日本語を使用する](#)」の項目を参照してください。

[4-5] アーティクル

Articles (アーティクル) はBead (ビーズ) と呼ばれる矩形エリアのタイトルとリストからなります。おのおのbeadはARTICLEの名前でpdfmark operatorを使用することが規定されています。beadsはそれらがPostScriptファイル中に現れる順番でarticleに追加されます。

Articles作成の記述は：

```
[ /Title String
  /Rect [llx lly rux ury]
  /Page pagenum
  /ARTICLE
pdfmark
```

Key	Type	意味
Title	string	(必須) 一つのbeadのArticleのタイトル。最大65,535文字。3 2文字以内が推奨される。キャラクタセットとエンコーディング方式はPDFDocEncodingを使用。
Rect	array	(必須) [Xll Yll Xur Yur]の4つの数値で記述。[Xll Yll Xur Yur]とは四角形で表示されるリンクボタン枠の左下(x, y)と右上(x, y)を表す。lower-left x, y upper-right x, y
Page	integer	(オプション) beadを配置するところのページ番号。bead markerはPageキーを設定している場合、PostScriptファイル中のどこに記載しても良い。Pageキーを使用しない場合は、PostScriptファイル中のbeadを表示するページに該当する範囲内に配置されなければなりません。

さらに、一つのアーティクル内の最初のbeadは独自のKey value pairsを規定されます。推奨するキーはSubject, Author及びkeywordです。

```
サンプルコード

% Article containing two beads
[ /Title (Now is the Time)
  /Author (John Doe)
  /Subject (Coming to the aid of your country)
  /Keywords (Time, Country, Aid)
  /Rect [ 225 500 535 705 ]
  /Page 2
  /ARTICLE
pdfmark

[ /Title (Now is the Time)
  /Rect [ 225 500 535 705 ]
  /Page 3
  /ARTICLE
pdfmark
```

解説

このコードは特に解説の必要はないと思います。ここで理解すべきことは、/Rectで指定された矩形エリアがアーティクルのエリアであり、ここにマウスポインタが入ると、アーティクルポインタへ変わりアー

ティクルが設定されていますよとユーザーに知らせる、クリックするとその矩形エリアが画面上に表示できる最大のサイズに自動拡大される。再度クリックすると、次のアーティクル (bead) で設定される矩形エリアを /Rect で規定されたサイズで表示します。共通のアーティクル名として、/Title (Now is the Time) が設定されています。このタイトル名を一つのアーティクルとします。つまり、このタイトルがついたbeadが一連のクリック動作で次々に画面上に表示されることになります。

このコードをDistillし、Exchangeでアーティクルのプロパティを見てみると以下のようにになっています。



The image shows a dialog box titled "アーティクルのプロパティ" (Article Properties). It contains four text input fields and two buttons. The fields are labeled as follows:

- タイトル: Now is the Time
- サブタイトル: Coming to the aid of your country
- 作成者: John Doe
- キーワード: Time, Country, Aid

At the bottom right, there are two buttons: "キャンセル" (Cancel) and "OK".

このコードは/Pageキーを使用している限り、PostScriptファイル内のどこに記述してもかまいません。

[4-6] Named destinations (アンカー)

しおりまたはリンクのジャンプ先はページ番号とViewの替わりに一つの名前として指定することもできる。アクロバットビューアがnamed destination (名前付けられたジャンプ先)に出会うと、名前と一致したページとViewを決めたnamed destinationのPDFファイルテーブルを見に行きページを表示します。named destinationを使用する優れた点は、特にクロスドキュメントリンクで、リンク元を含む書類からリンク先を含む書類の改訂において、リンク先が異なったページへ移動されたかどうかということに関心を払わないでいいということである。すなわち、ここで規定する名前でアンカーが埋め込まれるということである。named destinationはDESTの名前でpdfmark operatorを使用することが規定されている。

named destination マーカー作成の記述は：

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

pdfmark

Key	Type	意味
Dest	name	(必須) destinationの名前
Page	integer	(オプション) ジャンプ先のページ番号 このキーが使用されていると、named destination マーカーはPostScriptファイル中のどこに配置しても良い。このキーがない場合は、PostScriptファイル中のジャンプ先ページに該当する範囲内に配置されなければなりません
View	array	(オプション) ジャンプ先ページを表示するViewモード指定。もしないと、null destination (100%の拡大表示、オリジンは左下コーナー)に初期設定される。View destinationについては後述します。

さらに、named destinationは独自のKey value pairsを規定されます。

アクロバット3.0では、named destinationが<http://www.adobe.com/test.pdf#name>のようにキャラクタ"#"のあとにURLに付加することができます。Acrobatビューワはnamed destinationで指定されるPDFファイルの一部を表示します。つまり、HTMLで記述されるページから直接PDFファイルの指定されたページを或いは場所へリンクすることができる。

```
サンプルコード

[ /Dest /MyNamedDest
  /Page 3
  /View [ /FitH 5 ]
  /DEST
pdfmark
```

解説

3ページ目にMyNameDestの名前でnamed destinationを設定する。いわゆるアンカーネームを埋め込む。表示するモードはView [/FitH 5]で規定される。詳細はView destinationを参照下さい。

Webリンクの場合では、<http://www.adobe.com/test.pdf#MyNamedDest>というふうに記述されます。

[4-7] Page cropping (ページトリミング)

ページトリミング(ページの切り取り)はPAGES及びPAGEの名前でpdfmark operatorを使うことによって規定される。PAGESは書類すべてのページに対して適用されるページトリミングの初期設定を規定する。一方PAGEは現在のページのみに対してページトリミングが規定されます。

書類に対してのデフォルトページトリミングを規定する記述は：

```
[          /CropBox [l1x l1y urx ury]
/PAGES
pdfmark
```

このマーカはPostScriptファイル中の何処に置かれても良いが、ファイルの最初の方に配置することが推奨されます。最初の方とは、書類構築コメントの%%BeginSetupと%%EndSetupの間のDocument Setupセクションで、最初のページでほかのマーカが配置される前。Distillerがサムネールを作成するようセットアップするとき、このマーカがファイルの最初の方に配置されていない場合では、正しくないサムネールが作成されてしまうこともあります。

書類中のあるページにのみに対してページトリミングを規定する記述は：

```
[          /CropBox [l1x l1y urx ury]
/PAGE
pdfmark
```

このマーカはそのページに対して作用させるためにはshowpage operatorの前に配置されなければなりません。マーカの場所はページ上にほかのマーカが配置される前に配置されることが推奨されます。例えば、このマーカが最初のページの全てのマーカより前に配置されている場合では、書類の最初のページのみに作用します。

Key	Type	意味
CropBox	array	(必須) ページの表示エリアのサイズ及び場所。CropBoxは[X1 Y1 X2 Y2]の4つの数値で記述。[X1 Y1 X2 Y2]とは四角形で表示されるノートの左下(x, y)と右上(x, y)を表す。lower-left x, y upper-right x, y。詳細はここを参照ください。このマーカはPostScriptファイル中のプリント時に実行されるコードのブロックの場所に配置されなければなりません。

```
サンプルコード

% Crop all pages
[          /CropBox [54 403 558 720]
/PAGES
pdfmark

% Crop this page
[          /CropBox [0 0 612 792]
/PAGE
pdfmark
```



[4-8] Info Dictionary (書類情報)

書類のInfo dictionaryはその書類についての種々の情報であるところのkey-value pairsを持ちます。Info dictionary情報は、DOCINFOの名前でpdfmark operatorを使用するように規定されています。

Info dictionaryマーカー作成の記述は：

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

pdfmark

Key	Type	意味
Author	string	(オプション) 書類の所有者
CreationDate	string	(オプション) 書類の作成日、stringフォーマットについては、ModDateの項を参照
Creator	string	(オプション) 別のフォームからPDFへ変換された場合のオリジナル書類を作成したアプリケーションの名前。QuarkXpressやPageMaker等
Producer	string	(オプション) PDFへ変換したアプリケーションの名前。Acrobat Distiller3.0J等
Title	string	(オプション) 書類のファイル名
Subject	string	(オプション) 書類のサブジェクト名
Keywords	string	(オプション) 書類について関係するキーワード。書類間のサーチで使われる。現状の日本語版では使われない。
ModDate	string	(オプション) 書類の最終修正日時。以下のフォームとなる。 (D:YYMMDDHHmmSSOHH'mm') 前述の項目を参照。

Info dictionaryマーカーはPostScriptファイルのどこに配置してもかまいません。さらに、上記テーブルのキーに追加して、独自のキーを規定することもできます。値はストリングオブジェクトでなければなりません。

サンプルコード

```
[ /Title (My Test Document)
  /Author (John Doe)
  /Subject (pdfmark 2.0)
  /Keywords (pdfmark, example, test)
  /Creator (Hand Programmed)
  /ModificationDate (D:19940912205731)
  /ADBEstest_MyKey (My private information)
  /DOCINFO
```

pdfmark

ここは解説よりも実際どのようにこれらの情報が追加されるかをReaderで見てください。「ファイル」→「書類情報」より「一般」を選択すると、下のような書類情報ウィンドウが現れますので、確認してください。

一般情報

ファイル名: Macintosh HD:Desktop Folder:0.pdf

タイトル: My Test Document

サブタイトル: pdfmark 2.0

作成者: John Doe

キーワード: pdfmark, example, test

作成: Hand Programmed
PDF 変換: Acrobat Distiller 3.0J (Power Macintosh 版)

作成日時: 98.3.15 1:48:12 AM
更新日時: 98.3.15 1:48:12 AM

ファイルサイズ: 18917 バイト 最適化: いいえ

キャンセル OK

[4-9] 表示モード

書類のCatalog dictionaryはその書類についての種々の情報を備えるところのkey-value pairsを持ちます。以下のCatalog dictionary情報はpdfmark operatorを使用することでセットすることができます。

- ファイルを開いたときに発生するアクション。
- 書類を開いたときの表示方法。書類のみ、書類とサムネール、書類としおり、及び、フルスクリーンモードを選択できます。

Catalog dictionary作成の記述は：

```
[ /PageMode name
...Action-specifying key-value pairs...
/DOCVIEW
pdfmark
```

Key	Type	意味
PageMode	name	(オプション) 書類を開けたとき、どのように表示するかの指定。以下の一つでなければならぬ。 UseNone -- しおり、サムネールのどちらでも表示しなくて書類を開く。 UseOutlines -- しおりを表示して書類を開く。 UseThumbs -- サムネールを表示して書類を開く。 FullScreen -- フルスクリーンモードで書類を開く。 初期設定はUseNoneである。つまり、このマーカーを使用していなければ、書類のみが表示される。

上記テーブルに追加して、カタログはアクションを規定するadditional key-value parisを持つオープンアクションを含む。詳細は、後述するアクションの項目を参照。

Catalog dictionaryマーカーはPostScriptファイルのどの場所に配置されても良い。

```
サンプルコード

%--しおりモードで2ページ目表示
[ /PageMode /UseOutlines
  /Page 2 /View [/XYZ null null null]
/DOCVIEW
pdfmark

%--フルスクリーンモードで2ページ目を表示
[ /PageMode /FullScreen
  /Page 2
/DOCVIEW
pdfmark
```

解説

書類を開けたときの表示モードはUseOutlines、つまりしおり付きで表示する。 /Page 2 2ページ目を指定して、 /View [/XYZ null null null] で指定されるページ位置及び拡大率で表示する、この記述では現在の表示モードが敬承されます。 /Viewの記述については、後述のアクションの項を参照。



[4-10] トランジション

PDFファイルをプレゼンなどの資料として使用する場合に、ページ切り替え時のトランジション効果をPDFファイルに付加することが出来ます。ReaderやExchangeでは、フルスクリーンモード時のトランジションの設定を「ファイル」-「環境設定」-「フルスクリーンモード」で現れる環境設定ウインドウから設定することが可能ですが、フルスクリーンモード時しか利用できない、効果は1種類のみであるの制約があります。トランジションのpdfmarkを使用することで、この制約を無くし、どのモードでも、ページの章ごとや、毎ページでも色々な効果を付けるを可能にします。トランジションの記述は各ページに対するDictionaryにこの記述を追加することで実現します。pdfmark operatorは/PUTを使用します。

トランジションをDictionaryに追加する記述は：

```
{PageN} << /Trans << ... key value-pairs... >> >> /PUT pdfmark
```

または

```
{ThisPage} << /Trans << ... key value-pairs... >> >> /PUT pdfmark
```

Key	Type	意味
Type	name	(オプション) オブジェクトタイプ、常にTrans
S(Subtype)	name	(オプション) トランジション効果を説明。このキーを削除するとトランジション効果が得られない。下のDキーも無効になります。トランジション効果は以下のテーブルで説明します。
D(Duration)	real	(オプション) トランジション効果の時間(秒)出、初期設定は1秒です。

トランジション効果

プレゼンテーションモード(フルスクリーン)の全ての実行環境は以下のテーブルのトランジション効果をサポートします。これらの効果のいくつかは、効果の表示方法をコントロールするオプションパラメータを含んでいます。

効果	パラメータ	説明
Split	Dm, M	スプリット。2つのラインがスクリーンをスイープ。Dm(Dimension)キーはスイープラインの垂直(/V)、水平(/H)方向を指定する。M(Motion)キーは方向、内から外側(/O)、外から内側(/I)を指定する。
Blinds	Dm	ブラインド。複数のラインがスクリーンをスイープ。Dmキーはスイープラインの垂直(/V)、水平(/H)方向を指定する。
Box	M	ボックス。ボックスがセンターから外側、または、外側からセンタースクリーンをスイープする。Mキーは方向、内から外側(/O)、外から内側(/I)を指定する。
Wipe	Di	ワイプ。シングルラインがスクリーンをスイープする。Di(Direction)キーは角度を指定(0, 90, 180, 270)
Dissolve	none	ディゾルブ。パラパラと新しいページイメージが現れる。
Glitter	Di	グリッター。ディゾルブと同じ効果であるが、スイープの方向をDiキーで指定。サポートする方向は0, 270 315。
R	none	(Replace-置き換え)。効果は単純に古いイメージから新しいページイメージに置き換えられる。従い、トランジション効果はない。これは、もし、Sキーがトランジションディクショナリにない場合のデフォルト効果であるが、フルスクリーンモードに使用されるどの

デフォルトトランジション(Viewerの環境設定で指定される)に対して最優先する方法として規定される。

サンプルコード

```
[ {Page1} << /Trans << /S /Split /D 0.5 /M /I /Dm /V >> >>  
/PUT
```

pdfmark

```
[ {Page3} << /Trans << /S /Blinds /D 1.5 /Dm /H>> >>  
/PUT
```

pdfmark

```
[ {ThisPage} << /Trans << /S /Wipe /D 1 /Di 180 >> >>  
/PUT
```

pdfmark

```
[ {ThisPage} << /Trans << /S /Glitter /D 1.5 /Di 315 >> >>  
/PUT
```

pdfmark

[4-11] アディショナルアクション

書類のあるページを開いたとき、カーソルがある矩形範囲に入ったとき、クリックされたとき等に応じたアクションを指定することができます。これらの記述は{Catalog}ディクショナリで規定されます。書き込みはAA(Additional Action)キーを使用し、以下のモードがあります。

アディショナルアクションの記述は各ページに対するDictionaryにこの記述を追加することで実現します。pdfmark operatorは/PUTを使用します。

アディショナルアクションをDictionaryに追加する記述は：

```
[{PageN} << /AA << ... key value-pairs... >> >> /PUT pdfmark
```

または

```
[{ThisPage} << /AA << ... key value-pairs... >> >> /PUT pdfmark
```

アディショナルアクションの属性

Key	Type	意味
E(Enter)	dictionary	(オプション) カーソルがアクティベーションエリアに入ったときに実行されるアクション
X(Exit)	dictionary	(オプション) カーソルがアクティベーションエリアから出たときに実行されるアクション
D(Down)	dictionary	(オプション) マウスボタンがアクティベーションエリア内で押下されたときに実行されるアクション
U(Up)	dictionary	(オプション) マウスボタンがアクティベーションエリア内で押上げられたときに実行されるアクション
O(Open)	dictionary	(オプション、ページオブジェクト用として定義される)ページが表示された後に実行されるアクション
C(Close)	dictionary	(オプション、ページオブジェクト用として定義される)ページが切り替えられた後に実行されるアクション
FP(first page)	dictionary	(オプション)書類の最初のページが表示されたときに実行されるアクション
PP(prev. page)	dictionary	(オプション)前ページが表示されたときに実行されるアクション
NP(next page)	dictionary	(オプション)次ページが表示されたときに実行されるアクション
LP(last page)	dictionary	(オプション)書類の最終ページが表示されたときに実行されるアクション

これらのアクションを使用する場合には、予め実行するアクションが設定されていなければなりません。例えば、3ページ目が表示された場合にムービーが再生するようにするには、予め、3ページ目にムービーファイルが貼り付けられていなければなりません。その後、このキーで実行されるオブジェクトを指定します。dictionaryの記述は実行するアクションに従います。

例えば、書類を開くと“おいでやす”、閉じると“バイバ~イ！”。次ページを選択すると“次ページに行きます！”なんて言うサウンド効果を付けることもできますね。

サンプルコード

%--ページ3にムービー貼り付け

```
[ /Rect [150 150 450 350]
  /SrcPg 3                                %--ムービー貼り付けページ
  /Subtype /Movie
  /Title (space ship)
  /Movie << /F (space.mov) /Poster true>>
  /A <</Showcontrols false /Mode /Open /Volume 0.8 >>
  /F 1
  /Border [0 0 10[5]]
  /Color [0 1 0.5]
  /ANN
pdfmark
```

%--ページ1ムービー貼り付け

```
[ /Rect [150 100 450 280]
  /SrcPg 1                                %--ムービー貼り付けページ
  /Subtype /Movie
  /Title (ktworld)
  /Movie <</F (kt.mov) /Poster false >>
  /F 1
  /ANN
pdfmark
```

%--オープンアクションの設定-3ページのディクショナリにAdditional Action追加

```
[ {Page3} << /AA << /O << /S /Movie
  /T (space ship) /Operation /Play >> >> >>
  /PUT
```

pdfmark

%--クローズアクションの設定-1ページのディクショナリにAdditional Action追加

```
[ {Page1} << /AA << /C << /S /Movie
  /T (ktworld) /Operation /Play >> >> >>
  /PUT
```

pdfmark

[4-12] 初期表示設定

書類を開いたときの初期設定は{Catalog}ディクショナリで規定されます。書き込みはViewerPreferencesキーを使用し、以下のモードがあります。

初期表示設定の記述は書類に対するDictionaryにこの記述を追加することで実現します。pdfmark operatorは/PUTを使用します。

ViewerPreferenceをDictionaryに追加する記述は：

```
{Catalog}<</ViewerPreferences <<... key value-pairs... >> >>/PUT pdfmark
```

Key	Type	意味
HideToolbar	Boolean	(オプション) ツールバーの非表示
HideMenubar	Boolean	(オプション) ニューバーの非表示
HideWindowUI	Boolean	(オプション) ウィンドウコントロールの非表示 サムネール、しおり表示、ページ番号、拡大率などの情報を非表示にする。
FitWindow	Boolean	(オプション) ページサイズにウィンドウサイズを合わせる
CenterWindow	Boolean	(オプション) ウィンドウを中央に表示する
PageLayout	name	(オプション) SinglePage シングルページ表示 OneColumn 1コラム表示 TwoColumnLeft 2コラム表示で、1ページ目が左 TwoColumnRight 2コラム表示で、1ページ目が右
NonFullScreenPageMode	name	(オプション) PageMode Catalog中のPageModeがFullScreenに設定されている場合で、このモードを抜けた後のページモードを指定する。

```

サンプルコード

%--カタログディクショナリに初期表示設定を記述
%--ツールバー非表示で2コラムモードで1ページ目を右で表示
[
  {Catalog}<</ViewerPreferences<< /HideToolbar true /PageLayout /TwoColumnRight>> >>
  /PUT
pdfmark

%--表示ウィンドウを書類サイズに合わせ、スクリーンのセンターで表示。
%--ページ番号や拡大率などの情報は非表示
[
  {Catalog}<</ViewerPreferences<< /HideToolbar true /HideMenubar true /HideWindowUI true /FitWindow true /CenterWindow true /PageLayout /TwoColumnRight>> >>
  /PUT
pdfmark

%--フルスクリーンモードを抜けたらしおり表示モードにする
[
  {Catalog}<</ViewerPreferences <</NonFullScreenPageMode /UseOutlines>> >>
  /PUT
pdfmark

```

注) 組み合わせるキーによっては、相反する記述になったり、組み合わせることができない場合もあるので注意

が必要です。例えば、/DOCVIEWで/FullScreenを指定しているのに、ここの表示モードで/PageLayout /TwoColumnRightとやっても、/DOCVIEWの記述が優先されます。また、上記2番目のサンプルコードのように/HideToolbar true と/FitWindow trueを指定しても、仕様のためか/FitWindow trueしか有効になりません。

[4-13] フォームフィールドの作成

pdfmark operatorを使用してPDFファイルの任意のページにフォームフィールドを作成する方法を以下に記述します。

フォームフィールドの記述はリンクやノートのアノテーションマーカである/ANNを使用します。

フォームフィールド作成の記述は：

```
[ /Subtype /Widget ... key value-pairs... /ANN pdfmark
```

1999.02.18改正

Key	Type	意味
Subtype	name	(必須) 常にWidget (フォームフィールドアノテーション)
NeedAppearances	boolean	(オプション) true or false フォームフィールドの表示・非表示 1ページ上に複数個作成する場合には必須となる。
SrcPg	Integer	(オプション) 整数 フォームフィールドを作成するページ番号
Rect	Array	(必須) フォームフィールドの座標位置 [llx lly urx ury] (ポイント)
T(Title)	string	(必須) フォームフィールドの名前 英名であること
TU(UserName)	string	(オプション) フォームフィールドのユーザー名
FT	name	<p>(必須) フィールドタイプ Tx : テキストフィールド Btn : ボタンフィールド Ch : チョイスフィールド</p> <p>フィールドのタイプ、ボタン・コンボボックス・リストボックス・チェックボックス等を決定するのは、このFTキーと次で説明するFfキー(フィールドフラグ)の組み合わせで決まります。すなわち以下の通りになります・</p> <p>Button Field</p> <p>Ex: /FT /Btn /Ff 65540 プッシュボタン /FT /Btn チェックボックス /FT /Btn /Ff 32768 ラジオボタン</p> <p>Choice Fields</p> <p>/FT /Ch リストボックス /FT /Ch /Ff 131072 コンボボックス</p> <p>Text Field</p> <p>/FT /Tx テキスト</p>
Ff	number	<p>(必須) フィールドフラグ フォールドのタイプを指定</p> <p>1 bit 1: 上書き禁止 2 bit 2: 必須 4 bit 3: フィールド出力禁止 4096 bit 13: 複数行 8192 bit 14: パスワードフィールド 16384 bit 15: 初期選択をチェック(ラジオボタン・チェックボックス)</p> <p>32768 bit 16: ラジオボタン</p>

		65540 bit 17: プッシュボタン 131072 bit 18: コンボボックス 393216 bit 19: 編集可(コンボボックス) 524288 bit 20: アイテムのソート(リストボックス) 655360 bit : アイテムのソート(コンボボックス) 917504 bit : アイテムのソート・編集可(コンボボックス) Ex: /Ff 2 必須項目
H	name	(オプション) ハイライト P : プッシュ ハイライト I : 反転 O : アウトライン N : なし Ex: /H /I 反転
F	number	(オプション) フラグオプション Fキーなし : 印刷禁止 1: 印刷禁止 2: フィールド非表示 & 印刷禁止 4: 印刷可 6: フィールド非表示 Ex: /F 4 印刷可
BS	disctionary	(オプション) フィールドスタイル W : 境界線の太さ (1 2 3 ...) ポイント S : サブタイプ S : 実線 B : ベベル I : インセント U : アンダーライン D : 破線 D : 間隔 [array] ex. /D [3 2] Ex: /BS << /W 3 /S /B >> ボーダー太さ 3ポイント、ベベル /BS << /W 1 /S /D /D [3 2] >> ボーダー太さ 1ポイント、3 2 ポ イントの破線
MK	disctionary	(オプション) フィールドの表示属性 BC : 境界線色 [r g b] BCキー無しは透明 BG : 背景色 [r g b] BGキー無しは透明 CA : 通常時のボタン表示テキスト (string) または、チェックボックスの場合では (4) Check (l) Circle (8) Cross (u) Diamond (n) Square (H) Star #チェックボックスでの/DAキーは (/ZaDb 0 Tf 0 g) でZapf Dingbatsのシンボルフォントを使用する子こと。 RC : ロールオーバー時の表示テキスト (string) AC : ボタン押下時の表示テキスト (string) Ex: /MK << /BC [1 1 0] /BG [0.5 0 0] /CA(Normal Text) /RC(Rolled Over Text) /AC(Pushed Text) >> Ex: /MK << /BC [1 0 0] /BG [0 0 0] /CA(4) >> チェックマークシン ボルのチェックボックス
		(オプション-プッシュボタン) Default Appearance フォント属性 KaGo : フォント名 KaGo : 平成角ゴシック

DA	name	<p>KaMi : 平成明朝 TiRo : Times Roman Tilt : Times Italic TiBo : Times Bold TiBl : Times Bold Italic Helv : Helvetica HeOb : Helvetica Oblique HeBo : Helvetica Bold HeB0 : Helvetica Bold Oblique Cour : Courier CoOb : Courier Oblique CoBo : Courier Bold CoB0 : Courier Bold Oblique Symb : Symbol ZaDb : Zapf Dingbats</p> <p>12 : フォントサイズ 自動設定の場合は 0を指定 Tf 1 0 0 rg: フォントカラー (Tf red green blue rg) - RGBカラー指定 (1 1 1で白) (Tf 0.5 g) - グレースケール指定(1で白)</p> <p>Ex: /DA (/KaGo 12 Tf 1 0 0 rg) 平成角ゴシック12ポイント、赤 Ex: /DA (/ZaDb 0 Tf 1 g) Zapf Dingbats オートポイント、グレースケール黒</p>
AP	disctionary	<p>(オプション=プッシュボタン) Appearance カスタムボタンフィールドの表示属性</p> <p>N : 通常時のカスタムボタン表示 {XObject} R : ロールオーバー時のカスタムボタン表示 {XObject} D : ボタン押下時のカスタムボタン表示 {XObject}</p> <p>Ex: /AP << /N {XObjectName1} /R {XObjectName2} /D {XObjectName3} >></p> <p>APキーはカスタムでXObjectで指定された画像データを表示するもので、予め{XObject}で記述するPostScriptコードが必要です。</p>
AS	disctionary	<p>(オプション=プッシュボタン) Appearance State カスタムボタンフィールドのリモート表示</p>
V	name	<p>(オプション) Value リストボックス・コンボボックスのアイテムまたは出力値の初期値, および, テキストフィールドでの初期値</p> <p>Ex: /V /Yes /V /Checked Ex: /V (Value)</p>
DV	name	<p>(オプション) Default Value リセットされた場合の初期設定値</p> <p>Ex: /DV /C2 Ex: /DV (Default Value)</p>
\$On	name	<p>(オプション) ラジオボタン・チェックボックスでの出力値</p> <p>Ex: /\$On /Yes Ex: /\$On /Elephant</p>
Q	number	<p>(オプション=テキスト) テキストの整列</p> <p>0 : 左揃え 1 : 中央揃え 2 : 右揃え</p> <p>Ex: /Q 1 中央揃え</p>
MaxLen	number	<p>(オプション=テキスト) 最大入力文字数</p>

		Ex: /MaxLen 20 最大文字数20
Opt	array	<p>(オプション-リストボックス・コンボボックス) リストボックス・コンボボックスにリストする項目</p> <p>Ex: /Opt [(Lion)(Tiger)(Elephant)(Monkey)(Panda)]</p> <p>出力値を設定する場合は、アイテムの前に出力値を記述し以下のようなフォーマットになる。 [[value1)(item1)][value2)(item2)][value2)(item3)]・・・] となる。 Ex: /Opt [[(1)(Lion)][(2)(Tiger)][(3)(Elephant)][(4)(Monkey)][(5)(Panda)]</p>
A(Action)	dictionary	<p>(オプション) アクションの設定 詳細なアクションの設定はここを参照ください S (Subtype) /GoTo : GoToアクションの指定</p> <p>Ex: /A << /S /GoTo /D [{PageN} /XYZ null null null]>> Nページへのジャンプアクション このキーはDistillするとき実際のページが存在しない場合には無効になる。 Ex:/A << /S /JavaScript /JS (this.pageNum = 75 ;) >> 75+1ページへのジャンプ (JavaScript使用例) このキーでは実際のページが存在しなくても有効。後からページを挿入すれば、そのページにジャンプできる。</p> <p>フォームフィールドのリセットアクション Ex: /A << /S /ResetForm >> 全フィールドのリセット /A << /S /ResetForm /Fields [(FieldName1) (FieldName2)] >> 指定フィールドのリセット</p>

サンプルコード

%-- プッシュボタン ハイライト ベベル 平成角ゴシック クリックで4ページにジャンプ-----

```
[
  /Subtype /Widget
  /NeedAppearances true
  /SrcPg 1
  /Rect [100 200 350 250]
  /T (FieldName)
  /FT /Btn
  /Ff 65540
  /H /P
  /F 4
  /BS << /W 3 /S /B >>
  /MK << /BC [ 0 0 1 ] /BG [ 0 1 0 ] /CA (Normal Text) /RC (Roll Over Text)/AC (Pushed Text) >>
  /DA (/KaGo 12 Tf 1 0 0 rg)
  /A << /S /GoTo /D [ {Page4} /XYZ null null null ] >>
  /ANN
pdfmark
```

%-- テキストフィールド 平成明朝 初期設定テキスト(Default Text) 境界線一破線-----

```
[
  /Subtype /Widget
  /NeedAppearances true
  /SrcPg 1
  /Rect [100 300 350 350]
  /T (TextName)
  /FT /Tx
  /F 4
  /BS << /W 1 /S /D /D 3 >>
  /MK << /BC [ 0 0 0 ] /BG [ 0.7 0.7 0.7 ] /CA (Default Text) >>
  /DA (/KaMi 24 Tf 1 0 1 rg)
  /ANN
pdfmark
```

```
%-- テキストフィールド 初期設定テキスト(Default Text) -----  
[  
  /Subtype /Widget  
  /NeedAppearances true  
  /SrcPg 1  
  /Rect [100 650 300 700]  
  /T (TextName)  
  /FT /Tx %テキストフィールド  
  /Ff 2 %必須  
  /F 4  
  /V (Default Text)  
  /DV (Default Text)  
  /Q 0 %左揃え  
  /BS << /W 1 /S /I >>  
  /MK << /BC [ 0 0 0 ] /BG [ 1 1 0.8 ] >>  
  /DA (/TiBo 24 Tf 0 1 0 rg)  
  /ANN
```

pdfmark

```
%-- プッシュボタンフィールド ハイライト ベベル 平成角ゴシック 4ページ目にジャンプアクション  
[  
  /Subtype /Widget  
  /NeedAppearances true  
  /SrcPg 1  
  /Rect [100 550 300 600]  
  /T (PushBtnName)  
  /FT /Btn %プッシュボタン  
  /Ff 65540 %プッシュボタン  
  /H /P  
  /F 4  
  /BS << /W 3 /S /B >>  
  /MK << /BC [ 0 0 1 ] /BG [ 0 1 0 ] /CA (Normal Text) /RC (Roll Over Text)  
  /AC (Pushed Text) >>  
  /DA (/KaGo 24 Tf 1 0 0 rg)  
  /A << /S /GoTo /D [ {Page4} /XYZ null null null ] >>  
  /ANN
```

pdfmark

```
%-- リストボックス-----  
[  
  /Subtype /Widget  
  /NeedAppearances true  
  /SrcPg 1  
  /Rect [100 350 300 500]  
  /T (ListBoxName)  
  /FT /Ch %チョイスフィールド リストボックス  
  /Ff 524288 %ソートアイテム  
  /F 4  
  /BS <</S /I >>  
  /MK << /BC [ 1 0 0 ] /BG [ 1 0.8 0.8 ] >>  
  /DV (Elephant) %フィールドがリセットされたときの値  
  /V (Lion) %書類を開けたときの初期設定値  
  /Opt [(Lion)(Tiger)(Elephant)(Monkey)]  
  /DA (/Helv 24 Tf 1 0 1 rg)  
  /ANN
```

pdfmark

```
%-- コンボボックス-----  
[  
  /Subtype /Widget  
  /NeedAppearances true  
  /SrcPg 1  
  /Rect [100 250 300 300]  
  /T (ComboBoxName)  
  /FT /Ch %チョイスフィールド  
  /Ff 393216 %コンボボックス  
  /F 4  
  /BS <</S /I >>  
  /MK << /BC [ 0 0 0 ] /BG [0.8 0.8 1] >>  
  /DV (Tiger) %フィールドがリセットされたときの値
```

```
/V (Monkey)
/Opt [(Lion)(Tiger)(Elephant)(Monkey)]
/DA (/TiBo 0 Tf 0 0.5 0.5 rg)
/ANN
```

%書類を開けたときの初期設定値

pdfmark

%-- フォームのリセット-----

```
[
  /Subtype /Widget
  /NeedAppearances true
  /SrcPg 1
  /Rect [100 150 300 200]
  /T (ResetForm)
  /FT /Btn %プッシュボタン
  /Ff 65540 %プッシュボタン
  /H /P
  /F 4
  /BS << /W 1 /S /B>>
  /MK << /BC [ 1 0 0 ] /BG [ 0.5 0.5 0.5]
  /CA (Clear to Defaults) /AC (Boo!) >>
  /DA (/HeBo 24 Tf 0 0 1 rg)
  /A << /S /ResetForm >>
  /ANN
```

pdfmark

%-- チェックボックス-----

```
[
  /Subtype /Widget
  /NeedAppearances true
  /SrcPg 1
  /Rect [400 700 420 720]
  /T (CheckBox4-1)
  /FT /Btn %チェックボックス
  /F 4
  /BS <</W 1/S /S>>
  /MK << /BC [ 0 0 0 ] /CA (4)>> %チェックマーク
  /DA (/ZaDb 0 Tf 0 g)
  /$On /Lion %出力値
  /ANN
```

pdfmark

```
[
  /Subtype /Widget
  /NeedAppearances true
  /SrcPg 1
  /Rect [400 660 420 680]
  /T (CheckBox4-2)
  /FT /Btn %チェックボックス
  /F 4
  /BS <</W 1/S /S>>
  /MK << /BC [ 0 0 0 ] /CA (I)>> %円形マーク
  /DA (/ZaDb 0 Tf 0 g)
  /$On /Tiger %出力値
  /ANN
```

pdfmark

```
[
  /Subtype /Widget
  /NeedAppearances true
  /SrcPg 1
  /Rect [400 620 420 640]
  /T (CheckBox4-3)
  /FT /Btn %チェックボックス
  /F 4
  /BS <</W 1/S /S>>
  /MK << /BC [ 0 0 0 ] /CA (8)>> %十字形マーク
  /DA (/ZaDb 0 Tf 0 g)
```

/\$On /Elephant %出力値
/ANN

pdfmark

[

/Subtype /Widget
/NeedAppearances true
/SrcPg 1
/Rect [400 580 420 600]
/T (CheckBox4-4)
/FT /Btn %チェックボックス
/F 4
/BS <</W 1/S /S>>
/MK << /BC [0 0 0] /CA (u)>> %ダイヤモンドマーク
/DA (/ZaDb 0 Tf 0 g)
/\$On /Monkey %出力値
/ANN

pdfmark

[

/Subtype /Widget
/NeedAppearances true
/SrcPg 1
/Rect [400 540 420 560]
/T (CheckBox4-5)
/FT /Btn %チェックボックス
/F 4
/BS <</W 1/S /S>>
/MK << /BC [0 0 0] /CA (n)>> %四角形マーク
/DA (/ZaDb 0 Tf 0 g)
/\$On /Cat %出力値
/ANN

pdfmark

[

/Subtype /Widget
/NeedAppearances true
/SrcPg 1
/Rect [400 500 420 520]
/T (CheckBox4-6)
/FT /Btn %チェックボックス
/F 4
/BS <</W 1/S /S>>
/MK << /BC [0 0 0] /CA (H)>> %星マーク
/DA (/ZaDb 0 Tf 0 g)
/\$On /Other %出力値
/ANN

pdfmark

%-- ラジオボタン-----

[

/Subtype /Widget
/NeedAppearances true
/SrcPg 1
/Rect [500 700 520 720]
/T (RadioBtn-1)
/FT /Btn %チェックボックス
/Ff 32768 %ラジオボタン
/F 4
/BS <</W 1/S /S>>
/MK << /BC [0 0 0] /CA (I)>> %チェックマーク
/DA (/ZaDb 0 Tf 0 g)
/Parent {RadioBtn-1}
/\$On /Monkey %出力値
/ANN

pdfmark

[

/Subtype /Widget
/NeedAppearances true
/SrcPg 1

```
/Rect [500 660 520 680]
/T (RadioBtn-1)
/FT /Btn %チェックボックス
/Ff 32768 %ラジオボタン
/F 4
/BS <</W 1/S /S>>
/MK << /BC [ 0 0 0 ] /CA (I)>> %円形マーク
/DA (/ZaDb 0 Tf 0 g)
/Parent {RadioBtn-1}
/$On /Lion %出力値
/ANN
```

pdfmark

[

```
/Subtype /Widget
/NeedAppearances true
/SrcPg 1
/Rect [500 620 520 640]
/T (RadioBtn-1)
/FT /Btn %チェックボックス
/Ff 32768 %ラジオボタン
/F 4
/BS <</W 1/S /S>>
/MK << /BC [ 0 0 0 ] /CA (I)>> %十字形マーク
/DA (/ZaDb 0 Tf 0 g)
/Parent {RadioBtn-1}
/$On /Tiger %出力値
/ANN
```

pdfmark

[4-14] Pass-through PostScript commands

PostScript Language CodeをPDFファイル中に直接埋め込むことが/PSキーを使って可能です。この埋め込まれたPostScriptコードはDistill時には実行されること無く埋め込まれ、Acrobat ReaderやExchangeでこのPDFファイルを表示しても画面には表示されません。しかし、PostScriptプリンタに印刷出力する場合には、認識され出力されます。

これは電子透かしとして著作権の表示やメッセージを印刷することとして利用できます。

[サンプル例を見る](#)

Pass-through PostScript commandsを規定する記述は：

```
[ /DataSource string or file
  /Level1 string or file
  /PS
```

pdfmark

Key	Type	意味
DataSource	string or file	(必須) PDFファイルに埋め込むPostScript language コマンド
Level1	string or file	(オプション) PDFファイルに埋め込むPostScript language コマンド (PostScript Level1のみをサポートするプリンタで出力する場合に有効)

サンプルコード

```
%--ラインを表示
[ /DataSource (0 0 moveto 100 700 lineto stroke)
/PS
pdfmark

%--テキストの表示
[ /DataSource (/Jun501-Bold-RKSJ-H 20 selectfont
1 0.5 0.5 setrgbcolor 5 70 moveto (高品位印刷禁止) show)
/PS
pdfmark
```

[5] ディスティネーションとアクションの規定

PDFはユーザーがファイルを開けた時、リンクをクリックした時、および、しおりをクリックした時に何が
おこるかを規定する3つの方法をサポートします。3つの方法とは、view destinations, actions、およ
び、named destinatioです。

View destinationsはアcroバットの1.0と2.0のバージョンでサポートされ、同じファイル中の別の場所を
指定出来ます。

Actionsはアcroバットの2.0のバージョンでサポートされ、これらは、view destinationのコンセプトの一
般である。actionの4つのタイプがacrobat viewerのバージョン2.0でサポートされます。それらは；

- GoTo -- View destinationsと同じ機能。
- GoToR -- 別のPDFファイルの場所を規定する。
- Launch -- 独自のアプリケーション。または、書類を起動する。
- Article -- 指定されたarticleの読み出しを始める。

Named destinationsについては[前述を参照](#)ください。

Open actionを持っている書類のしおり、リンク、および、カタログディクショナリは下表のkeyリストの一
つを正確に含んでいなければなりません。3つの規定スキームが使用される場合によっては、さらにkey-
value pairsが必要とされます。これらは以下の段で説明します。

Key	Type	意味
Action	Name or Dictionary	<p>アクションのタイプを規定する。前定義されたname又はアクションディクショナリのどちらかでなければならない。 nameの場合、以下の一つでなければならない</p> <p>GoTo 現在の書類中で指定されたページやズームでジャンプする。Destキー、又は、Page , Viewキーを必要とする。詳細はGoTo actionsの項目参照。</p> <p>GoToR 別のPDF書類で指定されたページやズームでジャンプする。Destキー、又は、Page , Viewキー、及びfile-specifier keyを必要とする。詳細はGoToR actionsの項目参照。</p> <p>Launch 書類やアプリケーションを起動する。file-specifier keyを必要とする。詳細はLaunch actionsの項目参照。</p> <p>Article 現在の書類内又は別のPDF書類のアーティクルへジャンプする。Destキーを必要とする。さらに、別のPDFファイルのアーティクルの場合、file-specifier keyを必要とする。詳細はArticle actionsの項目参照。</p> <p>file-specifier keysのFile, MacFile, DOSFile, UnixFile, URI、及び、IDは外部ファイルを定義します。最低でもこれらのキーの一つはfile-specifier keyが必要とするところのアクションのために定義されなければならない。 もし、アクションの値がディクショナリの場合はカスタムアクションを規定します。カスタムアクションディクショナリはDistillerのバージョン2.1以上でサポートされ、一般に、Acrobat Exchangeプラグインアクションで取り扱うところのアクションである。詳細はCoustom actionsを参照。 アクションの初期設定値はGoToである。</p>
Dest	name, integer,	<p>アーティクルアクションのジャンプ先、又は、全てのアクションのnamed destination (アンカー)を規定する。named destinationの場合、値はDEST pdfmarkで定義されるジャンプ先の名前と一致するname objectでなければなりません。 Article destinationでは、値は書類中のアーティクルインデックスを指定する数</p>

	string	Article destinationには、但し言類下のアーティクルのリンク先を規定する数値、または、アーティクルのタイトル名に一致する文字列である。詳細はNamed destinationsを参照
View	array	ページ上のリンク、しおりのリンク先とfitタイプを規定する。

[5-1] View destinations

Viewキーの値はページの番号、ロケーション及びfitタイプを規定するarrayである。ロケーションは一つの矩形、ポイント、あるいは、x, y座標のいずれかであり、fitタイプに従う。

Key	Type	意味
Page	integer or name	ジャンプ先のページ。整数の値が指定されると、それは、PDFファイルのページ番号でなければならない。一つのファイルの最初のページは1である。0ではない。リンクとアーティクルのために、nameオブジェクトNextとPrevがジャンプ先のページの値として使用できる。もし、リンクのジャンプ先ページが同じページの場合、Pageキーは不要である。もし、Pageキーの値が0の場合、しおり、リンクはnull destinationを持つ。つまり、無効である。

Viewキーの値であるdestinationは下記テーブルのfitタイプ名とパラメーターの一つを含むarrayとして規定されます。

Fitタイプ名とパラメーター

Name	パラメーターと意味
Fit	パラメーター無し ページを表示ウインドウに合わす。これは、ページ全体を矩形とした場合のFitRで規定するショートカットである。ウインドウサイズを変えてもページ全体がウインドウ内で表示されるようページサイズが変更される。
FitB	パラメーター無し ページコンテンツのbounding boxを表示ウインドウに合わす。ウインドウサイズを変えても最初に表示されたページサイズを保持。
FitH	top ページの幅を表示ウインドウに合わす。topはページ原点（左下）からウインドウの上部への距離を規定する。ウインドウサイズを変えてもページ全体がウインドウ内で表示されるようページサイズが変更されるが、指定距離より小さいウインドウ高さではページサイズは固定。
FitBH	top ページコンテンツのbounding boxの幅を表示ウインドウに合わす。topはページ原点（左下）からウインドウの上部への距離を規定する。ウインドウサイズを変えてもページ全体がウインドウ内で表示されるようページサイズが変更されるが、指定距離より小さいウインドウ高さではページサイズは固定。
FitR	x1 y1 x2 y2 パラメータで規定される矩形を表示ウインドウに合わす。x1 y1=矩形の左下、x2 y2=矩形の右上の座標をポイントで指定。原点(0 0)はページの左下。
FitV	left ページの高さを表示ウインドウに合わす。leftはページ原点（左下）からウインドウの左部への距離を規定する。ウインドウサイズを変えてもページ全体がウインドウ内で表示されるようページサイズが変更されるが、指定距離より小さいウインドウ幅ではページサイズは固定。
FitBV	left ページコンテンツのbounding boxの高さを表示ウインドウに合わす。leftはページ原点（左下）からウインドウの左部への距離を規定する。ウインドウサイズを変えてもページ全体がウインドウ内で表示されるようページサイズが変更されるが、指定距離より小さいウインドウ幅ではページサイズは固定。
XYZ	left top zoom topとleftはページ原点（左下）からウインドウの左上部への距離を規定する。zoomは拡大率を規定し、1は100%の拡大である。もし、top, left, zoomがnullの場合、現在のパラメータを継承する。

例えば、/View [/null null null]では、現在のページと同じ水平、垂直オフセット及び拡大率を継承して表示される。zoom=0はnullと同じ意味です。

%%- サンプルコード

```
[ /Rect [70 650 210 675]
  /Page 3
  /View [/XYZ -5 797 1.5]
  /LNK
  pdfmark
```

[5-2] アクション

[5-2-1] GoToアクション

GoToアクションはview destinationと同じ情報を規定する。

[5-2-2] GoToRアクション

GoToRアクションは別のPDFファイル中のあるロケーションを規定する。これらはGoToアクションと同じ情報及びファイル名を持ちます。

Key	Type	意味
File	string	(必須) 起動するアプリケーションまたは書類のパス名、詳細はPDFレファレンスマニュアルを参照下さい。
URI	string	(オプション) PDFファイルのURL、Named destinationsはURLに"#"キャラクターを追加し、 http://www.adobe.com/test.pdf#name と言う記述を可能にする。アcroバットViewerはNamed destinationで指定されたPDFファイルの場所を表示する。

%%- サンプルコード

```
[ /Action /GoToR /File (test.pdf)
  /Page 2
  /View [/FitR 30 648 209 761]
  /Title (Open test.pdf on page 2)
  /OUT
pdfmark
```

[5-2-3] Launch アクション

Launchアクションは個別のアプリケーションまたは書類を起動する。

Key	Type	意味
File	string	(必須) 起動するアプリケーションまたは書類のパス名、詳細はPDFレファレンスマニュアルを参照下さい。
URI	string	(オプション) PDFファイルのURL、Named destinationsはURLに"#"キャラクターを追加し、http://www.adobe.com/test.pdf#nameと言う記述を可能にする。アクロバットViewerはNamed destinationで指定されたPDFファイルの場所を表示する。

```

%--サンプルコード
[
    /Rect [ 70 600 210 625 ]
    /Border [ 16 16 1 ]
    /Color [ 0 0 1 ]
    /Action /Launch
    /File (test.doc)
    /Subtype /Link
    /ANN
pdfmark

```

[5-2-4] Article アクション

アークティクルアクションはアクロバットViewerをアークティクル読み出しモードに設定する。

Key	Type	意味
Dest	integer or string	(必須) アークティクルアクションのリンク先、値は書類中のアークティクルインデックスを指定する整数。書類中の最初のアークティクルのインデックスは0である。または、アークティクルのタイトルに一致する文字列。
File	string	(必須) 起動するアプリケーションまたは書類のパス名、詳細はPDFレファレンスマニュアルを参照下さい。
URI	string	(オプション) PDFファイルのURL、Named destinationsはURLに"#"キャラクターを追加し、http://www.adobe.com/test.pdf#nameと言う記述を可能にする。アクロバットViewerはNamed destinationで指定されたPDFファイルの場所を表示する。

```

%--サンプルコード
[
    /Action /Article /Dest (Now is the Time)
    /Title (Now is the Time)
    /OUT
pdfmark

```

[5-2-5] カスタムアクション

カスタムアクションはあらかじめ定義された名前以外でのアクションタイプを作成することができます。

カスタムアクションはPDFファイルのアクションディクショナリに配置されたすべてのkey-value pairsを持つディクショナリによって規定されます。

1)URI アクション

World-Wide-Web上の場所へPDFファイルをリンクすることが出来るURIアクション。

```
/Action << /Subtype /URI /URI (http://www.adobe.com)
```

```
%-- サンプルコード
```

```
[  
    /Rect [ 50 425 295 445 ]  
    /Action << /Subtype /URI /URI (http://www.adobe.com) >>  
    /Border [ 0 0 2 ]  
    /Color [ .7 0 0 ]  
    /Subtype /Link  
    /ANN
```

```
pdfmark
```

2) Named アクション

Reader-Exchangeのメニュー項目を実行するNamedアクション。
この機能はExchangeのプラグインであるExcuteMenuItemを実現するものです。

```
/Action << /Subtype /Named /N /Print >>
```

メニューから「印刷」を選択したときと同じアクションを実行します。その他の項目は以下の表に示しません。

```
%-- サンプルコード(リンク)
```

```
[  
    /Rect [ 50 425 295 445 ]  
    /Action << /Subtype /Named /N /Print >>  
    /Border [ 0 0 2 ]  
    /Color [ .7 0 0 ]  
    /Subtype /Link  
    /ANN
```

```
pdfmark
```

```
%-- サンプルコード(フォームボタン 閉じる)
```

```
[  
    /Subtype /Widget  
    /NeedAppearances true  
    /SrcPg 1  
    /Rect [100 200 350 250]  
    /T (FieldName)  
    /FT /Btn  
    /Ff 65540  
    /H /P  
    /F 4  
    /BS << /W 3 /S /B >>  
    /MK << /BC [ 0 0 1 ] /BG [ 0 1 0 ] /CA (Close File) >>  
    /DA (/KaGo 0 Tf 1 0 0 rg)  
    /Action << /Subtype /Named /N /Close >>  
    /ANN
```

```
pdfmark
```

Named Keys

メニュー項目	メニューアイテムを実行するキー
ファイル	/Open, /Close, /Scan ¹ /Save ¹ , /SaveAs ¹ , /Optimizer:SaveAsOpt ¹ , /Print, /PageSetup, /Quit
ファイル 入力	/ImportImage ¹ /ImportNotes ¹ , /AcroForm:ImportFDF ¹
ファイル 出力 ¹	/ExportNotes ¹ , /AcroForm:ExportFDF ¹
ファイル 文書情報	/GeneralInfo, /OpenInfo, /FontsInfo, /SecurityInfo, /Weblink:Base ¹ , /AutoIndex:DocInfo ¹
ファイル 環境設定	/GeneralPrefs, /NotePrefs, /FullScreenPrefs, /Weblink:Prefs, /Acro-Search:Preferences (Windows) or /AcroSearch: Prefs (Mac) ² , /Cpt:Capture ¹
編集	/Undo, /Cut, /Copy, /Paste, /Clear, /SelectAll, /Ole:CopyFile ^{1,3} , /TouchUp: TextAttributes... ¹ , /TouchUp:FitTextToSelection ¹ , /TouchUp:ShowLineMarkers ¹ , /TouchUp:ShowCaptureSuspects ¹ , /TouchUp:FindSuspect ¹ , /Properties
編集 フィールド ¹	/AcroForm:Duplicate ¹ , /AcroForm:TabOrder ¹ Document ¹ /Cpt:CapturePages ¹ , /AcroForm:Actions ¹ , /CropPages ¹ , /Rotate-Pages ¹ , /InsertPages ¹ , /ExtractPages ¹ , /ReplacePages ¹ , /Delete-Pages ¹ , /NewBookmark ¹ , /SetBookmarkDest ¹ , /CreateAllThumbs ¹ , /DeleteAllThumbs ¹
表示	/ActualSize, /FitVisible, /FitWidth, /FitPage, /ZoomTo, /FullScreen, /FirstPage, /PrevPage, /NextPage, /LastPage, /GoToPage, /GoBack, /GoForward, /SinglePage, /OneColumn, /TwoColumns, /Article-Threads, /PageOnly, /ShowBookmarks, /ShowThumbs
ツール	/Hand, /ZoomIn, /ZoomOut, /SelectText, /SelectGraphics, /Note ¹ , /Link ¹ , /Thread ¹ , /AcroForm:Tool ¹ , /Acro_Movie:MoviePlayer ¹ , /TouchUp:TextTool ¹ , /Find, /FindAgain, /FindNextNote, /Create-NotesFile ¹
ツール 検索	/AcroSrch:Query, /AcroSrch:Indexes, /AcroSrch:Results, /AcroSrch:Assist, /AcroSrch:PrevDoc, /AcroSrch:PrevHit, /AcroSrch:NextHit, /AcroSrch:NextDoc
ウィンドウ	/ShowHideToolBar, /ShowHideMenuBar, /ShowHideClipboard, /Cascade, /TileHorizontal, /TileVertical, /CloseAll
ヘルプ	/HelpUserGuide ¹ , /HelpTutorial ¹ , /HelpExchange ¹ , /HelpScan ¹ , /HelpCapture ¹ , /HelpPDFWriter ¹ , /HelpDistiller ¹ , /HelpSearch ¹ , /HelpCatalog ¹ , /HelpReader ⁴ , /Weblink:Home
ヘルプ (Windows) or Appleメニュー (Mac)	/About

1 Exchangeのみ

2 プラットフォームにより独自に機能する

3 インストールされたOLEによりWindowsまたはMacで利用可能

4 Readerのみ

英語版のみサポート

3)JavaScript アクション

JavaScriptを実行するアクションの記述です。

```
/Action << /Subtype /JavaScript /JS (console.show¥(¥)¥r console.println¥("Cuckoo!"¥);) >>
```

```
%--サンプルコード(リンク)
[
  /Rect [ 100 100 400 200 ]
  /Action << /Subtype /JavaScript
  /JS (console.show¥(¥)¥r console.println¥("Cuckoo!";) >>
  /Subtype /Link
  /ANN
pdfmark
```

[5-2-6] Named destinations (アンカー)

Named destinationsはしおり、リンク、または、書類のカタログディクショナリのオプションOpen Actionのリンク先として用いることができます。

Key	Type	意味
Dest	name	リンク先名、DEST pdfmarkで定義したリンク先の名前に一致しなければなりません。

```
%--サンプルコード
[
  /Rect [ 70 650 210 675 ]
  /Border [ 16 16 1 [ 3 10 ] ]
  /Color [ 0 .7 1 ]
  /Dest /MyNamedDest
  /Subtype /Link
  /ANN
pdfmark
```

[6] pdfmarkで日本語を使用するには

pdfmarkを使用してノート、しおり名、書類情報、ア－ティクル名、日付文字列に日本語を記述することはそのままではできない。これらの文字列はUnicodeで取り扱われ、PDFDocEncodingという独自エンコード処理をした上で利用可能となります。

文字列

文字列は括弧"(", ")"によってデリミットされる。もし文字列が1行に収まらず長すぎる場合は、バックスラッシュ(日本語環境では"¥"円マーク、便宜的に以降の説明ではバックスラッシュとは"¥"を示します。)を使って複数行に分けることもできます。英文の場合以下ようになります。

```
( This is string number 1? )  
( strangeonium spectroscopy )  
( This string is split ¥  
across ¥  
three lines)
```

文字列でバックスラッシュキャラクター"¥"が括弧"(", ")"、Non-printable ASCIIキャラクター及びバックスラッシュ自身を使用するためにエスケープとして用いられます。このエスケープメカニズムはPostScript言語文字列と同じものです。詳細についてはPostScript Language Reference Manual, Second Editionを参照してください。

以下のテーブルはPDFのエスケープシーケンスリストです。

エスケープ処理	名称
¥n	linefeed
¥r	carriage return
¥t	horizontal tab
¥b	backspace
¥f	formfeed
¥¥	backslash
¥(left parenthesis
¥)	right parenthesis
¥ddd	character code ddd(octal)

¥ddd エスケープシーケンスの使用はPrintable ASCIIキャラクターセット範囲外のキャラクターを表すために最適な方法です。これは、キャラクターの送信や保存での潜在的な問題を最小にするためです。

つまり、文字コードがPrintable ASCIIキャラクターセット「20」から「7F」内にあれば上記例外処理(¥¥ backslash, ¥(left parenthesis, ¥) right parenthesis)を除いてそのままのキャラクターが使用される。これ以外の範囲では同様に制御コード(¥n linefeed, ¥r carriage return, ¥t horizontal tab, ¥b backspace, ¥f formfeed, ¥¥ backslash)を除いて¥ddd(8進コード)形式が適用されることとなります。参考として欧文コードテーブルを以下のように示します。



日本語の取り扱い

PDFバージョン1.1以降では、printable 7-bit ASCIIキャラクターセットのみを使用して文字列を表示する必要はありません。上記のようにしてnon-printable ASCIIコードを文字列として表わしたり、実際8-bitまでの文字を同様に文字列として表すことが可能だからです。

PDF中の情報(しおり、ノート、書類情報、ア－ティクル、日付)はUnicodeで表されます。PDF 1.1では、これらの文字列はISO Latin1の上位セットであるPDFDocEncodingで保存されます。PDFDocEncodingはUnicodeと256文字部分で互換性があります。

文字列はPDFDocEncoding又はUnicodeのどちらかで保存されます。もし、Unicodeであれば、文字列の最初の2バイトはUnicode Byte Orderマーカ－である<FE, FF>でなければなりません。<FE, FF>とは8進表示で<376, 377>になります。

つまり、Unicodeの文字列を表示するためにはその文字列の最初にこのUnicode Byte Orderマーカ－を挿入してこの文字はUnicodeでエンコードされていますと表示させなければなりません。さらにそれらはエスケープシーケンス処理されていなければなりません。つまり、PDF中に日本語文字を使用するためには、(¥376¥377 xxxxxxxxxxxxxxxx)という記述方法が採用されなければなりません。

では上記文字列の(xxxxxxxxxxxxxxxx)部分の記述方法は具体的にはどうなるのでしょうか。そのメカニズムを検証します。

例えば「あいうえお」をエンコーディングすると、どうなるのでしょうか。
「あ」はUnicodeで「3042」hexで表されますね、最初の「30」はPrintable ASCIIコードですのでASCII文字に直すと(0)です。次の「42」は(B)です。つまり「あ」は(0B)で表示されます。「い」はUnicodeで「3044」hexですから、「30」は(0)、「44」は(D)で「い」は(0D)で表示されます。同様に、「う」－(0F)、「え」－(0H)、「お」－(0J)、続けると、(0B0D0F0H0J)になります。

では「アイウエオ」の場合ではどうでしょう。
「ア」はUnicodeで「30A2」hex「30」は上記と同じく(0)、「A2」はPrintable ASCIIコードの範囲を超えますので、¥ddd(8進数)の形式でエスケープシーケンス処理され(¥242)になります。よって「ア」は(0¥242)で表示されます。

「イ」はUnicodeで「30A4」hex「30」は「0」、「A4」は「¥244」oct、「イ」は(0¥244)
「ウ」はUnicodeで「30A6」hex「30」は「0」、hex「A4」は「¥246」oct、「ウ」は(0¥246)
「エ」はUnicodeで「30A8」hex「30」は「0」、hex「A4」は「¥250」oct、「エ」は(0¥250)
「オ」はUnicodeで「30AA」hex「30」は「0」、hex「A4」は「¥252」oct、「オ」は(0¥252)
よって文字列は(0¥2420¥2440¥2460¥2500¥252)となります。

全角の「12345」では、
「1」はUnicodeで「FF11」hex「FF」は(¥377)oct、「11」は(¥021)oct、「1」は(¥377¥021)
「2」はUnicodeで「FF12」hex「FF」は(¥377)oct、「12」は(¥022)oct、「2」は(¥377¥022)
「3」はUnicodeで「FF13」hex「FF」は(¥377)oct、「13」は(¥023)oct、「3」は(¥377¥023)
「4」はUnicodeで「FF14」hex「FF」は(¥377)oct、「15」は(¥024)oct、「4」は(¥377¥024)

「5」はUnicodeで「FF15」hex「FF」は(¥377)oct、「15」は(¥025)oct、「5」は(¥377¥025)文字列は(¥377¥021¥377¥022¥377¥023¥377¥024¥377¥025)となります。

「松井秀喜」の場合では、
「松」はUnicodeで「677E」hex「67」は(g)、「7E」は(~)で「松」は(g~)
「井」はUnicodeで「4E95」hex「4E」は(N)、「95」は(¥225)で「井」は(N¥225)
「秀」はUnicodeで「79C0」hex「79」は(y)、「C0」は(¥300)で「秀」は(y¥300)
「喜」はUnicodeで「559C」hex「55」は(U)、「9C」は(¥234)で「喜」は(U¥234)
よって文字列は(g~N¥225y¥300U¥234)で表されます。

1バイト数字やアルファベットの場合では、Unicodeでの最初の1バイトは全て「00」ですので、null文字(¥000)で表され、その後1バイト文字が続きます。すなわち、「Acrobat3.0J」では(¥000A¥000c¥000r¥000o¥000b¥000a¥000t¥0003¥000.¥0000¥000J)で表されます。

これらの文字をPDFで使用する場合は、前述したようにUnicode Byte Orderマーカである(¥376¥377)に続けて配置し、以下のように記述されます。

「あいうえお」 (¥376¥3770B0D0F0H0J)
「アイウエオ」 (¥376¥3770¥2420¥2440¥2460¥250)
「12345」 (¥376¥377¥377¥021¥377¥022¥377¥023¥377¥024¥377¥025)
「松井秀喜」 (¥376¥377g~N¥225y¥300U¥234)
「Acrobat3.0J」 (¥376¥377¥000A¥000c¥000r¥000o¥000b¥000a¥000t¥0003¥000.¥0000¥000J)

pdfmarkで日本語を使用するには

例えば、pdfmarkで日本語を使ったしおりを作成する場合は、以下のような記述になります。

```
[/Count 2 /Page 1 /View [/XYZ 44 730 1.0]
  /Title (¥376¥3770B0D0F0H0J)
  /OUT pdfmark
[/Action /Launch /File (abcdef.doc)
  /Title (¥376¥3770¥2420¥2440¥2460¥250)
  /OUT pdfmark
[/Action /GoToR /File (abcdef.pdf) /Page 2 /View [/FitR 30 648 209 761]
  /Title (¥376¥377¥377¥021¥377¥022¥377¥023¥377¥024¥377¥025)
  /OUT pdfmark
[/Action /GoToR /File (abcdef.pdf) /Page 2 /View [/FitR 30 648 209 761]
  /Title (¥376¥377g~N¥225y¥300U¥234)
  /OUT pdfmark
[/Action /GoToR /File (abcdef.pdf) /Page 2 /View [/FitR 30 648 209 761]
  /Title (¥376¥377¥000A¥000c¥000r¥000o¥000b¥000a¥000t¥0003¥000.¥0000¥000J)
  /OUT pdfmark
```

これをDistillerで開きPDFに変換し、Exchangeでしおりを表示させると以下ようになります。



[7] RunDir, RunFileコマンドとの連携

QuarkXPressやPageMakerなどで1冊の本や書類を作成する場合、ページ数が数十から100ページ近くになる場合や、項目で区切られる場合などでは、章単位・ページ単位に分けてファイル作成されることが多いと思います。

また、Illustrator等のドローイングソフトで数個のイラスト等を作成する場合でも、1画像=1ファイルで作成されるのが一般的でしょう。

このように個別で作成されたファイルから1冊のPDFファイルにまとめるには、それぞれのファイルをPDFファイルに一度変換し、Exchangeを使用して、ページの挿入・移動の機能でページ組み立てます。

Acrobat Distillerにはこの作業を効率的に行なえるように'[RunFile"コマンド](#)'と'[RunDir"コマンド](#)'が用意されています。RunFileコマンドはPSファイルを指定してDistillするもので、RunDirコマンドはPSファイルが置かれているディレクトリを指定して、その中に含まれるPSファイルを全てDistillするものです。

さらにこれらのコマンドと併用してpdfmark operatorを使用することで、飛躍的にPDFファイルの加工の効率化が進みます。

- [RunFileコマンド](#)
- [RunDirコマンド](#)
- [pdfmarkを利用する](#)

[7-1] RunFileコマンド

これは、個別に書き出されたPSファイルの在り処（絶対パス記述）を、このRunFileコマンドに追加して記述したテキストファイルを作成し（拡張子は.ps）、Distillerに読み込ませる（開く）だけで、自動的にPSファイル群をかき集め順次Distillし、一つのPDFファイルとして生成します。

それでは、具体的にどうするのかを見てみましょう！

1. 個別に書き出されたPSファイル群を用意し、適当なフォルダに保存しておきます。（一個のフォルダにまとめる必要はありませんが、パス指定を記述するのが面倒なので今回はまとめます。また、同じ理由から階梯

~~本編は~~ ~~ま~~ ~~先~~ ~~で~~ ~~他~~ ~~の~~ ~~ド~~ ~~ク~~ ~~を~~ ~~指~~ ~~定~~ ~~し~~ ~~て~~ ~~記~~ ~~述~~ ~~す~~ ~~の~~ ~~が~~ ~~面~~ ~~倒~~ ~~な~~ ~~の~~ ~~で~~ ~~今~~ ~~回~~ ~~は~~ ~~ま~~ ~~と~~ ~~め~~ ~~ま~~ ~~す~~ ~~。~~ ~~ま~~ ~~た~~ ~~、~~ ~~同~~ ~~じ~~ ~~理~~ ~~由~~ ~~か~~ ~~ら~~ ~~階~~ ~~梯~~

```
(Macintosh HD:Folder:Index.ps) prun
```

```
% Windowsなら(c:/Folder/*.ps) prun *は実際のファイル名  
% になりますので上を書き換えて下さい。また、ハードディスク名が違う場合も有りますので  
% 使用になるPCの環境に合わせてパスを書き換えて下さい。
```

3. 作成したテキストファイルを1冊にまとめるPDFファイルの名前で保存します。拡張子は".ps"とする。例えば、"TestRunFile.ps"のようにします。保存先は特に指定は有りません。
4. Distillerを起動し、この"TestRunFile.ps"テキストファイルを読み込みます。生成するPDFファイル名を入力し「OK」を押すと、"TestRunFile.ps"内で書かれた順番でDistillが始まります。

[7-2] RunDirコマンド

これは、個別に書き出されたPSファイルを一つのフォルダにまとめておき、以下のRunDirコマンドに追加して記述したテキストファイルを作成し（拡張子は.txt）、Distillerに読み込ませる（開く）だけで、自動的に記述されたフォルダ内のPSファイル群を順次Distillし、一つのPDFファイルとして生成します。

それでは、具体的にどうするのかを見てみましょう！

1. 個別に書き出されたPSファイル群を用意し、適当なフォルダにまとめておきます。今回はMacintosh HDの直下にFolderの名前でフォルダを作成し、以下のPSファイルをおきます。
2. 以下のテキストをコピー＆ペーストしてテキストファイルを作成します。先頭に%がある文はコメント文ですので、実際は不要です。先頭の%!は必ず必要です。テキスト中、**シアンで表示された部分**を自分が作成したフォルダのパス名に書き換えます。（詳細はAcrobat 3.0Jフォルダ内のXtrasフォルダ内の"RunDirEx.txt"を参照下さい。）

```
%!  
/PathName (Macintosh HD:Folder:*.ps) def % Edit this to point to the folder  
% containing the PS files.  
/RunDir { % Uses PathName variable on the operand stack  
  { /mysave save def % Performs a save before running the PS file  
  dup = flush % Shows name of PS file being run  
  RunFile % Calls built in Distiller procedure  
  clear cleardictstack % Cleans up after PS file  
  mysave restore % Restores save level  
  }  
  255 string  
  filenameforall  
} def  
  
PathName RunDir  
  
% Windowsなら(c:/Folder/*.ps)  
% になりますので上を書き換えて下さい。また、ハードディスク名が違う場合も有りますので  
% 使用になるPCの環境に合わせてパスを書き換えて下さい。
```

3. 作成したテキストファイルを1冊にまとめるPDFファイルの名前で保存します。拡張子は".txt"とする。例えば、"TestRunDir.txt"のようにします。

注意) 拡張子を .ps とし先ほどのフォルダ内に配置しないで下さい。Distiller が 2 度同じ PS ファイル読み込んで PDF ファイルを生成してしまいます。

4. Distiller を起動し、この "TestRunDir.txt" テキストファイルを読み込みます。
生成する PDF ファイル名を指定し「OK」を押すと、指定されたフォルダ内の PS ファイルの Distill が始まります。
ここで注意すべきポイントはページ構成の順番は PS ファイル名のアルファベット順ですので、ファイル名がページ順になるような工夫が必要です。

[7-3] pdfmark operator を利用する

上述した "RunFile コマンド" ・ "RunDir コマンド" を pdfmark operator と合わせて使用することで、大変効率良く PDF ファイルを作成することができます。

pdfmark operator とは簡単にいうと Exchange で行う作業 (しおり作成、リンク、書類情報、ノート作成など) のほとんどはこのマーカー言語で記述することができ、これを PS ファイルに配置し Distill することでほぼ完成した PDF ファイルを作成できることです。

以下のような pdfmarker ファイルを作成し、テキストとして保存します。拡張子は .ps です。今回は "pdfmrkSsn.ps" として作成しました。

```
%!PS-Adobe-3.0
%%BeginProlog
/bd {bind def} bind def
/fsd {findfont exch scalefont def} bd
/sms {setfont moveto show} bd
/ms {moveto show} bd
systemdict /pdfmark known not
{userdict /pdfmark systemdict /cleartomark get put } if
%%EndProlog

%%BeginSetup

% OPEN INFO
[
    /PageMode /UseOutlines
    /Page 1 /View [/XYZ 44 730 2.0]
    /DOCVIEW
pdfmark

% CATALOG INFO
[
    /Title (Season Pictures)
    /Author (Koji Tama)
    /Subject (Season)
    /Keywords (season, calender, icon)
    /Creator (Hand Programmed)
    /ModificationDate (D:19980618)
    /MyKey (My season picture)
    /DOCINFO
pdfmark
</TITLE>

% VIEWER PREFERENCES
[ {Catalog}
    <</ViewerPreferences
    <</FitWindow true /CenterWindow true>> >>
    /PUT pdfmark

% BOOKMARKS
[/Count -8 /Page 1 /View [/XYZ 44 730 2.0] /Title (Season-1) /OUT pdfmark
[/Page 1 /View [/XYZ 44 730 2.0] /Title (Jan) /OUT pdfmark
[/Page 2 /View [/XYZ 44 730 2.0] /Title (May) /OUT pdfmark
[/Page 3 /View [/XYZ 44 730 2.0] /Title (Jun) /OUT pdfmark
```

```
[/Page 4 /View [/XYZ 44 730 2.0] /Title (Jul) /OUT pdfmark
[/Page 5 /View [/XYZ 44 730 2.0] /Title (Aug) /OUT pdfmark
[/Page 6 /View [/XYZ 44 730 2.0] /Title (Sep) /OUT pdfmark
[/Page 7 /View [/XYZ 44 730 2.0] /Title (Nov) /OUT pdfmark
[/Page 8 /View [/XYZ 44 730 2.0] /Title (Dec) /OUT pdfmark

[/Count 8 /Page 9 /View [/XYZ 44 730 2.0] /Title (Season-2) /OUT pdfmark
[/Page 9 /View [/XYZ 44 730 2.0] /Title (Jan) /OUT pdfmark
[/Page 10 /View [/XYZ 44 730 2.0] /Title (May) /OUT pdfmark
[/Page 11 /View [/XYZ 44 730 2.0] /Title (Jun) /OUT pdfmark
[/Page 12 /View [/XYZ 44 730 2.0] /Title (Jul) /OUT pdfmark
[/Page 13 /View [/XYZ 44 730 2.0] /Title (Aug) /OUT pdfmark
[/Page 14 /View [/XYZ 44 730 2.0] /Title (Sep) /OUT pdfmark
[/Page 15 /View [/XYZ 44 730 2.0] /Title (Nov) /OUT pdfmark
[/Page 16 /View [/XYZ 44 730 2.0] /Title (Dec) /OUT pdfmark
```

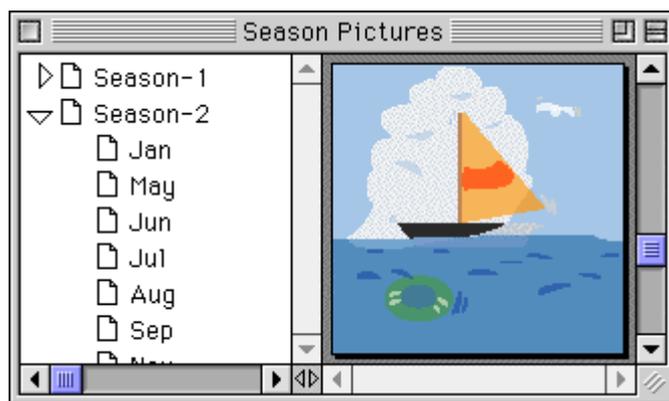
%%EOF

この記述を簡単に説明すると、

1. 書類を開いた時の表示スタイルは、しおり付きで表示、拡大率200%、また、ウインドウは書類サイズにフィットし画面のセンター位置で表示させる。
2. しおりはSeason-1, Season-2の名前で階層を持ち、各々(Jan.....Dec)の名前でページにリンクする。
3. 書類情報として、上記該当項目を入力する。

作成した"pdfmrkSsn.ps"ファイルを上記"RunFileコマンド"で作成したフォルダに入れ、同様に"RunFileSsn.ps"ファイルに"pdfmrkSsn.ps"ファイルのパス(下のパス)を記入し保存します。
(Macintosh HD:Folder:pdfmrkSsn.ps) prun

これを"RunFileコマンド"の場合と同様な手順でDistillerで読み込みます。作成されたRunFileSsn.pdfファイルをExchangeやReaderで開けると、以下のように表示されます。



書類情報を確認すると以下のように設定されています。

一般情報

ファイル名: MacDTP1b:PDFfiles:SsnPicts.pdf

タイトル: Season Pictures

サブタイトル: Season

作成者: Koji Tama

キーワード: season, calender, icon

作成: Hand Programmed
PDF 変換: Acrobat Distiller 3.0.2J (Power Macintosh 版)
作成日時: 98.6.23 4:50:49 PM
更新日時: 98.6.23 4:54:09 PM

ファイルサイズ: 393770 バイト 最適化: はい

キャンセル OK

勿論、この手法は[RunDirコマンド](#)でも使用できます。
