PDF利用は配布から双方向へ

2009年8月
アンテナハウス株式会社
http://www.antenna.co.jp
小林　徳滋
koba@antenna.co.jp

PDFの特長

- PDFは、従来、紙で行われていた情報伝達や管理を電子ファイル方式に置き換えるものである
- PDFは紙の特性である、用紙サイズ（寸法）をもち、その上に多様なデザインの文字や図形綺麗にレイアウトして印刷する方式をそのまま電子的に再現できるのが特長である
- 紙を基盤とする印刷文化を継承する点で、画面表示目的を主として用紙サイズのないWebページ（HTML）や画像形式などと異なる

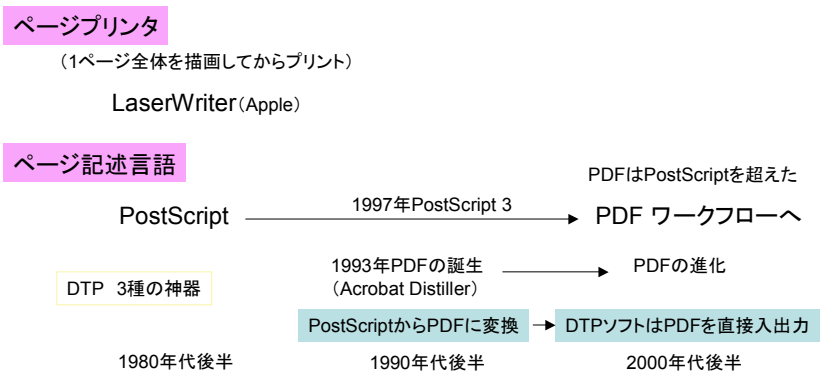
PDFの作成方法

PDFを作成する主な方法は次の4つである

① ページ記述言語（Postscript）から変換する
② Windowsアプリケーションの印刷機能を使って作成
③ アプリケーションから直接PDFファイルを作成
④ 書類をスキャナや複合機でスキャンしたデータをPDFに変換する

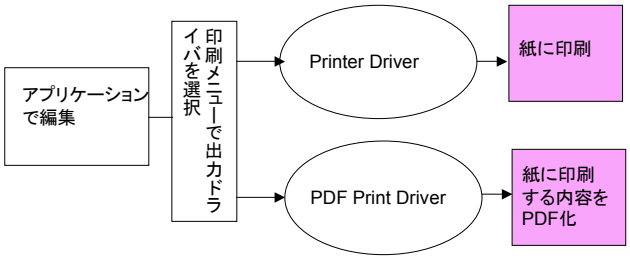
①PDFの起源はページ記述言語

1980年代半ば、DTPソフトでページを制作して、ページプリンタで印刷するDTP時代が始まった。
ページ記述言語PostScriptによる印刷工程がアナログ印刷工程に取って代わる。
情報を紙に表現するのと同等に電子的に配布するPDFが誕生した。



②印刷機能を使ってPDFを作る

アプリケーションがWindowsの表示・印刷機能（GDIという）を利用して、紙に出力する仕組みを利用して、紙に出す内容をPDFに出力する。



③ページを直接PDFに描画する

- ページの内容を直接PDFに出す。
- Windows GDIを使わないので、その制約を受けない。-->DTPソフトは概ねこの方式
  - 任意のカラースペースを使える
  - 解像度に依存しない
  - 印刷機能では出せない内容をPDFに出すことができる
- 高速である
- LinuxなどWindows以外でもPDFを作れる
- サーバサイドPDFに向いている

④電子化文書としてのPDF

- 紙の書類（書面）をスキャナーで読み取り画像化してその画像をPDFに変換するのは印刷起源のPDFとは別のものである
- 紙PDFにすることで、複数ページの管理、メタデータをつけての管理が簡単になる
- カラーのスキャナではファイルのサイズが大きくなるので、文字情報と画像情報を分離してそれぞれ適切な方式で圧縮してからPDF変換する「高解像度圧縮」が行われることが多い。
- PDFを全文検索できるように透明テキスト付きPDFにすることも多い。
  - OCRで認識したテキストを画像の上に配置し、テキスト検索可能にしたもの

PDFの種類

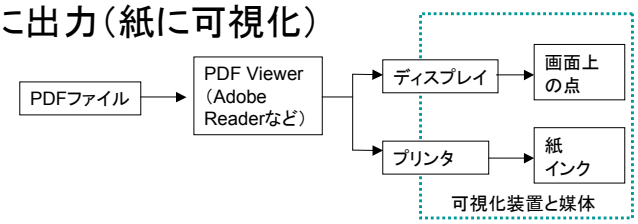
- PDFは二つの種類に大別できる
  - 電子文書としてのPDF
    - ①印刷技術から
    - ②アプリケーション・ソフトからPDFを出力
    - ③PDFの描画命令を直接記述する方法
  - 電子化文書としてのPDF
    - ④主に書類をスキャナで読み、そのデータをPDFに変換する
- 電子文書としてのPDFと電子化文書としてのPDFは全く異なる特性をもつ。
  - デジタル生まれのPDFはベクトル・データ（中心）
  - 紙から生まれたPDFはイメージ

画像ファイルをPDF化するメリット

- 前述のように、スキャナや複合機で読み取った画像をPDF化することが増えてきた。
- この理由は、
  - PDFの閲覧ソフトは非常に普及していることに加えて、複数のページの管理、画像に透過属性を与えることで、「高圧縮」、「透明テキスト」などが可能なことがある。
  - このような利用では、PDFファイル形式をコンテナ用として使っていると言える。

PDFの可視化

- PDF Readerは、PDFファイル内容中の描画命令を実行してファイルを可視化する
- 出力媒体
  - PCや携帯端末の画面
  - プリンタに出力（紙に可視化）



PDF批判に答える

- PDFは非常に普及した形式であるが、今も概ね、次のような批判をする人がいる。
- PDFは画面で読みにくい
  - PDFは用紙サイズの概念のある紙を画面に表示するのだが、ディスプレイのサイズとは異なるため読みにくい。
  - これは、サイズの概念をもつ用紙をモデルにする以上避けることができない。
- PDFは表示に時間がかかる
  - 一方、Adobe Readerはバージョン6位までかなり遅かったが、バージョン7で起動時間が1/3になり、また、バージョン9は8の1/2になるなど大きく改善された。

PDFの閲覧ソフト

- PDFを画面に表示する閲覧ソフトは、Adobe Readerが主たる存在である。
  - 他にも閲覧ソフトが存在する。
  - 閲覧ソフトは無償配布になっているためビジネスとしては成り立ちにくい。
- PDF活用をするには閲覧ソフトが肝になる。
  - Adobe Readerを使って、Adobeの競合製品を作ることができない。
  - 著作権管理などを行うには、Adobe Readerは使えない。
- 今後はさらに様々な閲覧ソフトが出てくる

PDFの仕様

- PDFの仕様はPDF Referenceとしてアドビが公開してきた。
- PDF ReferenceはAcrobatのバージョンアップと同期して、機能追加されて改訂されてきた。～1．7まで
- 2008年にAIIMに寄贈され、AIIMよりISOに提案されて標準化活動が行われた
- 2008年7月にISO 32000-1として発刊された
  - PDF Reference 1.7ベース

PDFファイル構造

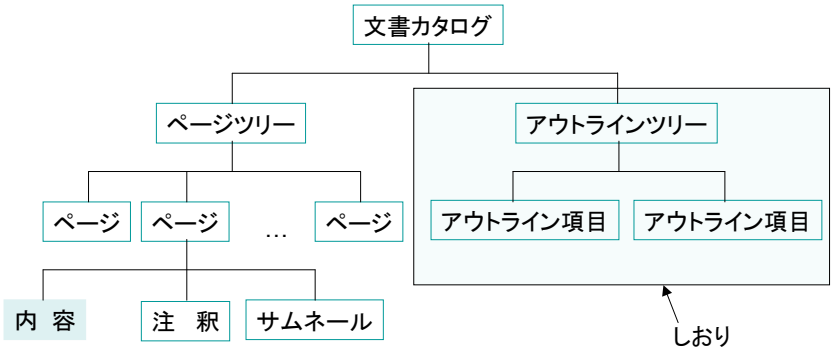
作成直後のPDF

ヘッダ　例）*%PDF–1.5*	PDFであることを識別するための情報
本体	PDFの本体情報
相互参照表	PDFの本体にランダムアクセスするための情報
トレイラ	PDFファイルは最後にファイルサイズ、カタログ情報、暗号辞書などが登録されている

標準ではトレイラが最後にあるため、Adobe Readerなどの利用アプリケーションは通常、PDFファイルの一番後ろから読まねばならない。

このため、ファイルの容量が大きい（ページ数の多い）PDFをWeb経由で表示しようとすると、全部ダウンロードするまで、画面には、内容がまったく表示できません。
【オプション】Web表示用に最適化（リニアライズ）で、ランダムアクセス用の情報を先頭に複製

PDFの構造（抜粋）

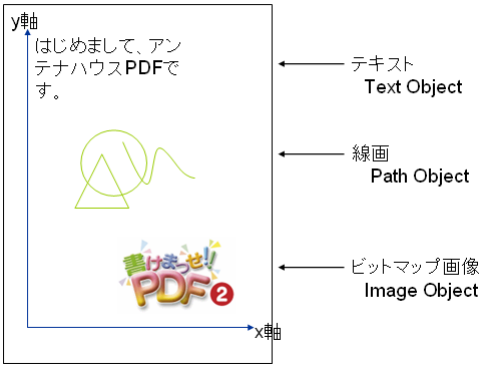


①PDFの内容は頁単位になっている（ワープロ文書（例：Word）などとは異質）
②しおりの情報は、アウトラインツリーとして別管理
③注釈は1頁毎に管理されていて、かつ、頁の内容とは別管理

PDFのページ内容

- PDFには1頁毎にページの内容を描画するための情報が保存されています。

PDFの内容表示用の主なオブジェクト

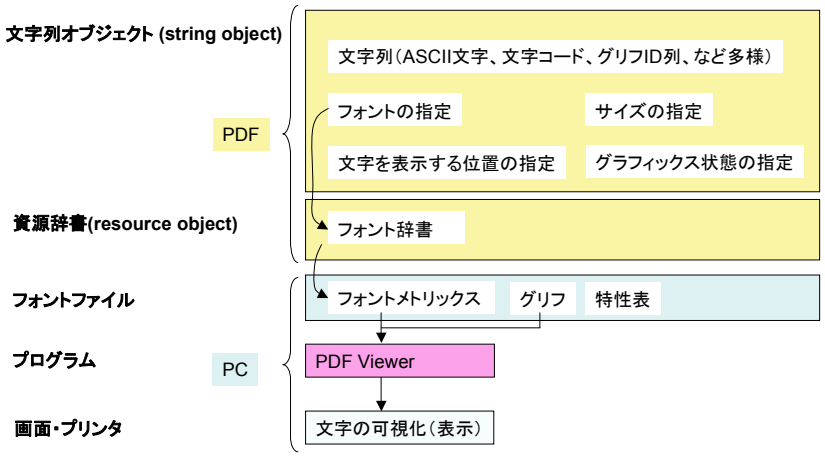


線画オブジェクトは2次元座標系の上に数学的な直線・曲線（パス）として表現されることができます。そうしたパスに線幅指定、色指定したり、パスで囲む領域を塗り潰したりすることで、図形が表現されます。

文字はビットマップとしてドットの塗り潰しで表すか、あるいは、文字の輪郭（アウトライン）を曲線で表して囲まれた部分を塗り潰すアウトラインフォントの方式で表現します。

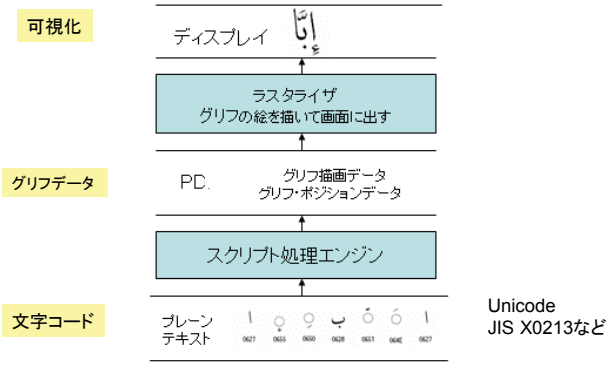
PDFファイルは、オブジェクトを規定するデジタルデータの塊

PDFでの文字表示



文字コードの可視化

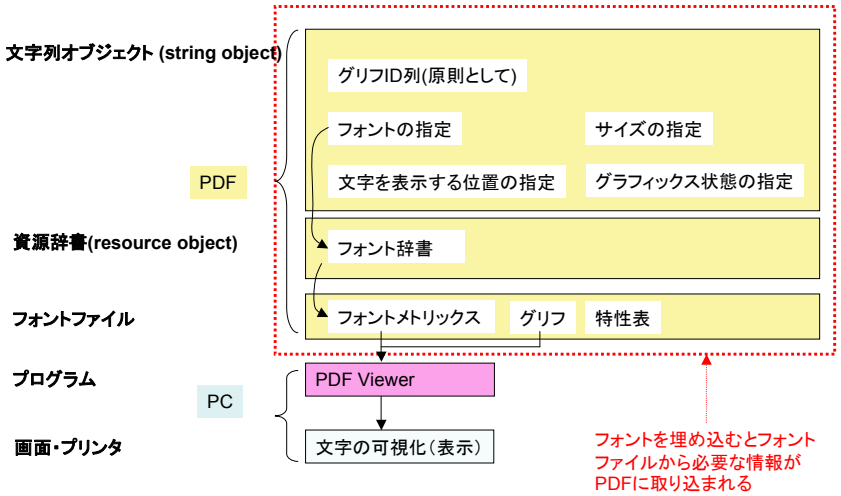
- 文字コードの並びは、フォントのグリフデータを用いて可視化される。



フォント埋め込み

- PDFを正しく表示できない、文字が化けるトラブルは大抵フォントを埋め込まないのが原因である。
  - 受信相手のシステム上のフォントを使う。
  - Adobe Readerは必ずしもそうなっていない。
  - 特に海外へ送るときは注意が必要。
    - 例）日本語の文字が少しでも入っていると英語のAdobe Readerではまったく表示できない。Windows¥Fontsにフォントがあってもだめ。
- フォントを埋め込んだPDF
  - PDFにフォントのサブセットが添付される。全文字ではなく使っている文字だけが原則。
  - TrueTypeやOpenTypeフォントには埋め込み可否のフラグがある。PDF作成ソフトはそのフラグをみて埋め込み処理をする。

フォントを埋め込んだPDF



イメージの扱い

- PDF Ref 1.7 Section 4.8 Images で定義
  - image Xobject（4.8.4）と inline-image (4.8.6)がある
- 4.8.4 Image Dictionaries
  - イメージの基本的属性を辞書で定義
    - 縦横のピクセル数
    - ピクセル単位のビット数
    - カラースペース
    - 圧縮方法(フィルタ)
    - 透過色

PDF Reference 1.7 pp343-344より引用

例:
幅256×高さ256、8ビットのイメージ
カラー空間:DeviceRGB
ページの左下(45,240)の位置に配置され
ユーザ空間の単位132の幅と高さに
スケーリングされて配置される

```
Example 4.28
20 0 obj % Page object
<< /Type /Page
/Parent 1 0 R
/Resources 21 0 R
/MediaBox [ 0 0 612 792 ]
/Contents 23 0 R
>>
endobj
21 0 obj% Resource dictionary for page
<< /ProcSet [ /PDF /ImageB ]
/XObject << /Im1 22 0 R >>
>>
endobj
```

右上へ続く

```
22 0 obj% Image XObject
<< /Type /XObject
/Subtype /Image
/Width 256
/Height 256
/ColorSpace /DeviceGray
/BitsPerComponent 8
/Length 83183
/Filter /ASCII85Decode
>>
stream
9LhZI9h¥GY9i+bb;,p:e;G9SP92/)X9MJ>^:f14d;,U(X8P
;cO;G9e];c$=k9Mn¥]
… Image data representing 65,536 samples …
8P;cO;G9e];c$=k9Mn¥]~>
endstream
Endobj

23 0 obj% Contents of page
<< /Length 56 >>
stream
Q % Save graphics state
132 0 0 132 45 140 cm % Translate to (45,140)
and scale by 132
/Im1 Do % Paint image
Q % Restore graphics state
endstream
endobj
```

イメージの実体データ

カラースペース（カラー空間）

- PDFは印刷技術から発展しただけに、カラー空間のサポートが充実している。
- PDFがサポートするカラー空間
  - CIEベース：CalRGB,CalGray, Lab, ICCBased
  - Deviceカラー：DeviceRGB, DeviceGray, DeviceCMYK
  - 特色：Separation, DeviceN, Indexed, Pattern

PDFのプロパティ

- メタデータ（基本の他、XMP形式のメタデータを組み込むことができる)
- タグ付きPDF
  - 音声読み上げのためにPDFに構造情報を埋め込む
  - アクセシビリティ
- Web表示に最適化（リニアライズ）
  - デフォルトのPDFは、インデックスがファイルの最後にあるため、最後までないと表示できない
  - リニアライズすると、インデックスがファイルの先頭にもできる

PDFのセキュリティ

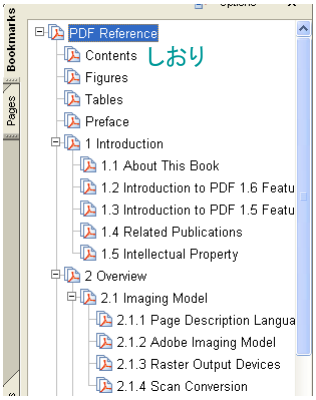
- 標準セキュリティハンドラが規定されており、Adobe Readerに備わっている。
- 方式
  - パスワード・セキュリティ
    - 閲覧制限パスワード（ユーザパスワード）
    - 編集制限パスワード（オーナーパスワード）
  - 公開鍵暗号方式セキュリティ
    - 公開鍵を使って暗号化。秘密鍵保有者のみ閲覧可。
- アルゴリズム
  - RC4：40ビット、128ビット
  - AES暗号：128ビット、256ビット

PDF閲覧時のナビゲーション機能

- 画面での閲覧をスムースにするために次のような機能が使える。
  - サムネイル
  - しおり
  - リンク

PDFのしおり

- しおり（Bookmark、アウトライン項目）
- しおりを階層化したツリーがアウトライン
  - 文書構造を表示する目次になる
- ドキュメント・カタログにて本体のページとは別に管理される



アウトライン

PDFのリンク

- PDFには次のようなリンクを付加することができる。
  - PDF内から外部PDFやWebページなどへのリンク
  - PDF内部同士のリンク
  - リンクにはアクションを指定することができる。

JavaScript

- Acrobat、Adobe Readerには、PDF独自拡張のJavaScriptインタープリタ（ランタイムライブラリー）が内蔵されている。
- フォルダにJSを追加することでAdobe Readerの動作を変更したり、PDFの中にJSを埋め込んでPDFを閲覧時の動作を変更、あるいはフォーム・フィールドにJSを付加することでフォーム入力時の動作をプログラムできる。

PDF/A長期保存

- 30年、50年を経過しても内容を元のまま閲覧できること
- ISO PDF/A：長期保存するための仕様
  - PDF/Aに準拠したPDFファイル生成は難しい
- 欧州が最も熱心である。
- 詳細は別スライド
  - PDFAOutline.pdf

PDFの双方向機能

- PDFを単に情報配布だけに使うのではなく、受け手がコメントを追加して送り手に戻す。
- 送り手は予め様式（フォーム）を規定しておく。受け手がその様式にデータを記入することで、送り手が受け手の情報を収集する。
  - 申請書などは、様式を変更されることなく、所定の箇所だけに記入して欲しい。

注釈（コメント）など

- 配布したPDFに注釈を付加
  - 双方向性
  - コメント注釈は付加情報で本文と別に管理される
- 注釈の種類
  - Text annotation
  - Link annotation
  - Free text annotation
  - Line annotation
  - Widget annotation （フィールドの概観）
  - . . .

アクロ・フォーム

- ユーザ対話データを保管
- 対話フォーム辞書
  - フィールド辞書を規定
  - フィールド辞書のタイプは次の4つ
    - ボタン・フィールド
    - テキスト・フィールド
    - 選択肢フィールド
    - 署名フィールド

→ 電子署名のデータを保管

PDFと電子署名

- 電子署名をセキュリティではなく、主に署名後に変更されていないことを検証するためのもの
- 署名フィールドを使う。
  - 未署名の署名フィールド
  - 署名済みの署名フィールド
- PDFの中に電子署名のデータを取り込む仕様である。
  - 署名済みPDFの署名データ（ハッシュ値）は署名フィールドの中の署名辞書に保管される。
- 署名の外観を付加できる。
  - 署名には、Widget注釈機能を使って、外観を添えることができる。概観のない署名（不可視署名）も可能。
- 署名を検証した結果を外観に表示できる。

PDF電子署名の特長

- 普通署名と証明用署名（MDP）という2種類がある。
- MDP署名は、PDFに付ける最初の書名であり、書名後のPDFに次の制限を課す。
  - 変更を許可しない
  - フォームフィールドの入力と書名フィールドに署名を許可
  - 注釈の作成、フォームフィールドの入力と書名フィールドに署名を許可
- 署名後のPDFは増分更新する。
  - 署名を次々に追加していくことも可能。
  - 署名バージョンという考え

PDFの長期署名

- 現在、標準化活動中
- 長期署名をPDFの仕様で実現する

PDFと全文検索

- 全文検索は文字コードがないとできない。
- 作成時に文字コードをPDF内に確保する。
- 画像からPDFを作成する場合には、画像をOCR処理して、文字コードをPDFに付加する。
  - 透明テキスト付PDF

PDFを紙のように活用する

- PDFは配布形式
  - PDFを変更することは原則として想定していない。
  - 本文の変更・修正の余地は小さい
    - タッチアップ（僅かな手入れ）が原則
- PDFで申請用紙等様式を配布することが多い。
  - 入力・申し込みを期待する。
  - 書式は変更せず、申請内容を追記する。
  - 紙に印刷して記入するのではなく、直接書き込みしたいのが人情。