

PDFファイルを自動生成

1. 1. はじめに

PDF ファイル ([Adobe Acrobat Reader](#)などで見えるファイルです) は、一般に広く利用されています。例えば、あなたがある企画書を Microsoft Power Point 2000 で書いたとして、その形式をそのままにクライアントに紙ではなく電子書式で送りたいと考えたとしましょう。そのとき、あなたは、「クライアントはMicrosoft Power Point 2000 を持っているかな?」と考えることでしょうか。現在、様々なプレゼンテーション用アプリケーション、表計算アプリケーション、ワープロアプリケーション、などのアプリケーションがあり、その数は数千、数万に及ぶことでしょう。Microsoft製品ならば、(少なくとも) あまり意識せずともよいこともあります。新しいバージョンのアプリケーションの場合など、やはり不安になることは多いはず。

PDFファイルは、そんな不安を解消してくれる共通書式を提供してくれます。PDFファイルに変換されたドキュメントは、Acrobat Readerを持っているクライアントに対して適切な表現方法を提供してくれます。

尚、このドキュメントはWindowsでの使用を前提にしていますが、Linux等であっても、Java系の方であれば、[「Jakartaの杜」](#)の[POI](#)を参照するのも良いでしょう。あるいは、[Apache FOP](#)というやり方もあります。

2. 2. 方法論

PDFファイルを作りたいと思ったら、PS (Post Script)形式からPDFに・・・と考えるのが本筋なのですが、てっとりばやくなんとかしたい人のために、PerlとVBS (VB Script) を使って、あるディレクトリにあるファイル群を一括してPDF形式に変換する方法をここでご提示いたします。勿論、一括して・・・などと考えない方は、Office2000製品がはいっている状態で、AcrobatをInstallするとPDFのアイコンができますので、簡単にPDFを作成できます。

まず、前提条件は、Adobe Acrobat 5.0 (日本語版) をインストールしておく必要があります。Adobe社のAcrobatは、インターネット黎明期からその地位を確立しており、現在、標準のフォーマットを提供している、といっても過言ではありません。

ません。ちなみに、Adobe Acrobat 5.0 は、13,000円くらいだったはずですが（訂正情報ください）。

Adobe Acrobat 5.0 には、Adobe Acrobat Writer というものと、Adobe Acrobat Reader というものと、Adobe Distiller というものが入っています。今後使っていくのは、Adobe Distiller です。ちなみに、Adobe Acrobat 4.05をWindows 2000をお使いの場合は、色々と問題が生じますので、Adobe社の情報を参照ください。また、色々と高度な事をしたい方は、Adobe社からSDKを入手し、そちらをご利用になることをおすすめします。

まず、Adobe Acrobat 5.0 をインストールした後に、Office のファイル（例：Microsoft Word 2000）で「印刷」をしたときにプリンタ名で「Adobe Acrobat Writer」と「Adobe Distiller」があるかどうかを確認ください。なければ以下は意味ありません。

あと、一応、以下のドキュメントの「LPT1:」の部分は、適宜お使いのOfficeのSP等により変わることもあります。はじめて行う際は、何かテスト用のエクセルファイルなりワードファイルなどをつくって、マクロで、Adobe Distiller にて印刷を行ってみて、そのマクロの内容をVBAで確認することで適宜変えてみてください。

以下、Microsoft Wordが前提です。Excel、PowerPoint等でも、VBAが存在するし、殆ど同じ手法を使えるのですが、Excel以外はまだ確認していないので、確認したらまたご報告します。

3. 3. VB Script (WSHやVBA) を使う方法

----wordpdf.vbs----

```
Set WordObj = WScript.CreateObject("Word.Application")
MyDoc = "C:\My Documents\test.doc" 'PDF化したいファイル
On Error Resume Next ' これやっとなないと、あとでエラーになったとき余計なプロセスが生じてしまう
WordObj.Documents.Open(MyDoc)
WordObj.Visible = False
currentPrinter = WordObj.Application.ActivePrinter
WordObj.ActivePrinter = "Acrobat Distiller on LPT1:" ' 環境に応じて書き換えてください。マクロを利用してこの部分を調べておくと良いです
WordObj.Options.UpdateFieldsAtPrint = True
WordObj.Options.PrintBackground = False
```

PDFファイルを自動生成

```
WordObj.Options.PrintReverse = True
WordObj.ActiveDocument.PrintOut
WordObj.ActiveDocument.Close
WordObj.Application.ActivePrinter = currentPrinter
WordObj.Quit
Set WordObj = Nothing
----word2pdf.vbs---- (フォルダ内のすべての.docファイルをPDF化)

Set WordObj = WScript.CreateObject("Word.Application")
WordObj.Visible = False
Path = "C:\TEMP" ' 検索フォルダ名の指定
Set FS = CreateObject("Scripting.FileSystemObject")
Set FD = FS.GetFolder(Path)
For Each File In FD.Files
    If Right(File.Name,3) = "doc" then ' Wordファイルであれば
        call word2pdf(Path & "\" & File.Name, WordObj) ' サブルーチンのコール
    end if
Next
Set FD = Nothing
Set FS = Nothing
WordObj.Quit
Set WordObj = Nothing
Sub word2pdf(filename, WordObj) ' 印刷用のサブルーチン
On Error Resume Next
WordObj.Documents.Open(filename)
WordObj.ActivePrinter = "Acrobat Distiller on Ne01:" ' 各自の環境にあわせて
WordObj.Options.UpdateFieldsAtPrint = False
WordObj.Options.PrintBackground = True
WordObj.Options.PrintReverse = False
WordObj.PrintOut
WordObj.Documents.Close
if Err.Number <> 0 then ' On error resume next しているので、エラーがおきてもエラーをトラップ可能
msgbox("error" & Err.Number & vbCR & filename & vbCR & Err.Description)
end if
End Sub
```

4. 4. Perlを使う方法

```

use strict;
use Win32::OLE;
use Win32::OLE::Const 'Microsoft Word';
my $CurrentPrinter;
my $MyDoc = "C:\\My Documents\\test.doc"; #ドキュメントの場所... ただし、\
は\\にする
my $Word = Win32::OLE->new('Word.Application', 'Quit');
my $const = Win32::OLE::Const->Load($Word);
$CurrentPrinter = $Word->{ActivePrinter};
$Word->{Visible} = 0;
$Word->{PrintReverse} => 1;
$Word->Documents->Open($MyDoc) || die("Unable to open document ",
Win32::OLE->LastError());
$Word->{ActivePrinter} = "Acrobat Distiller on LPT1:";
$Word->Options({UpdateFieldsAtPrint => 1,
PrintBackground => 0,
PrintReverse => 1,});
$Word->ActiveDocument->PrintOut({
Background => 0,
Append => 0,
Range => $const->{wdPrintAllDocument},
Item => $const->{wdPrintDocumentContent},
Copies => 1,
Collate => 0,
PageType => $const->{wdPrintAllPages}
});
$Word->{ActivePrinter} = $CurrentPrinter;
exit;

```

5. 5. EXCEL+VBScript(WSH)

----excelpdf.vbs---- (sheet1のみを印刷)

```

Set ExcelObj = WScript.CreateObject("Excel.Application")
MyDoc = "C:\\test.xls" ' PDF化したいファイル

```

PDFファイルを自動生成

```
On Error Resume Next ' これやっとなないと、あとでエラーになったとき余計なプロセスが生じてしまう
ExcelObj.Workbooks.Open(MyDoc)
ExcelObj.Visible = False
currentPrinter = ExcelObj.ActivePrinter
ExcelObj.ActivePrinter = "Acrobat Distiller on Ne01:" ' この部分は、各自の環境にあわせて
ExcelObj.Worksheets("Sheet1").Activate ' Sheet1のみActivate
ExcelObj.ActiveSheet.PrintOut ' Sheet1のみ印刷
ExcelObj.Workbooks.Close
ExcelObj.ActivePrinter = currentPrinter ' ここでエラーがでることがあります
. . .
ExcelObj.Quit
Set ExcelObj = Nothing
----excelpdf2.vbs---- (ブック全体を印刷)

Set ExcelObj = WScript.CreateObject("Excel.Application")
MyDoc = "C:\test.xls" ' PDF化したいファイル
On Error Resume Next
On Error Resume Next ' これやっとなないと、あとでエラーになったとき余計なプロセスが生じてしまう
ExcelObj.Workbooks.Open(MyDoc)
ExcelObj.Visible = False
currentPrinter = ExcelObj.ActivePrinter
ExcelObj.ActivePrinter = "Acrobat Distiller on Ne01:" ' この部分は、各自の環境にあわせて
ExcelObj.Copies = 1
ExcelObj.Collate = True
ExcelObj.ActiveWorkbook.PrintOut ' ブック全体を印刷：各SheetのPDF印刷のプロパティが一致していないと、最後のシートのみがPDF化されることがある
ExcelObj.Workbooks.Close
ExcelObj.ActivePrinter = currentPrinter ' ここでエラーがでることがあります
. . .
ExcelObj.Quit
Set ExcelObj = Nothing
----excel2pdf.vbs---- (フォルダ内の全エクセルファイルのブック全体を印刷)
```

```

Set ExcelObj = WScript.CreateObject("Excel.Application")
Path = "C:\temp" ' 検索フォルダ名の指定
Set FS = CreateObject("Scripting.FileSystemObject")
Set FD = FS.GetFolder(Path)
For Each File In FD.Files
  If Right(File.Name,3) = "xls" then ' エクセルファイルであれば...
    Call excel2pdf(Path & "\" & File.Name, ExcelObj) ' サブルーチンのCALL
  End If
Next
Set FD = Nothing
Set FS = Nothing
Set ExcelObj = Nothing
Sub excel2pdf(filename, ExcelObj) ' PDF印刷を行うファイルに対する処理のため
  のサブルーチン
  On Error Resume Next
  ExcelObj.Workbooks.Open(filename)
  ExcelObj.Visible = False
  ExcelObj.ActivePrinter = "Acrobat Distiller on Ne01:"
  ExcelObj.Copies = 1
  ExcelObj.Collate = True
  ExcelObj.ActiveWorkbook.PrintOut ' ブック全体を印刷：各SheetのPDF印刷のプ
  ロパティが一致していないと、最後のシートのみがPDF化される
  if Err.Number <> 0 then ' On error resume next しているので、エラーがおき
  てもエラーをトラップ可能
  msgbox("error" & Err.Number & vbCR & filename & vbCR & Err.Description)
  end if
  ExcelObj.Workbooks.Close
  ExcelObj.Quit
End Sub
---- (ここまで) ----

```

ちなみに、PDFの一ページに一枚のシートを印刷したい、などという場合については省略しますが、ここらへんは、EXCELマクロを記録してVBAのコード（Distillerを利用した印刷のすべてを記録）を参照する事で実装可能かと思います。

6. 6. まとめ

・・・すいません。本当は、もっと正確に書きたいのですが、時間がないもので・

PDFファイルを自動生成

・・・一応、成功したら、[こちら](#)からご一報ください。

C:\Program Files\Adobe\Acrobat 5.0\Distillr\PDF Output\
あるいは

C:\Program Files\Adobe\Acrobat 5.0\PDF Output\
(あるいは、ログインユーザーのデスクトップ)あたりに、ご希望のPDFファイルが

できていると成功です。

このフォルダも、実は変更可能です。コントロールパネルの「プリンタの設定」のPDF Distillerのプロパティって、他とちょっと違いますか？というか、フォルダ名+*.pdfがはいっていませんか？

例えば、C:\PDF\ディレクトリを、デフォルトの出力ディレクトリにしたければ、「プリンタの設定」で「Acrobat Distiller」を選び、プロパティを出します（マウスの左クリック）。プロパティシートの「ポート」で「新しいポートの作成」で<PDF Port>を選択→エクスプローラー風にフォルダ選択できる画面がでるので、C:\PDF\をさす・・・で、OKなはず（Windows2000, Acrobat5.0でのみ確認）。

あとですね。今回は、「自動化」させるといっても、高々1つのファイルなのでよいのですが、複数であれば、やはり、いちいちPDFファイルをつくったらPDFファイルがでてくるのってイヤですよ（あくまでサイレントモードでやりたい・・・）？そういう場合は、「プリンタの設定」で「Acrobat Distiller」を選び、プロパティを出します（マウスの左クリック）。プロパティシートの「全般」で、「印刷設定」ボタンを押します。「Adobe PDF 設定」タブにて（アイコンが入っているタブ）、「Acrobat でPDFを表示」と「PDFファイルの保存先を確認」の二つのチェックをはずします。以上でできるはず（Windows2000+Adobe Acrobat 5.0にてのみ確認）

あ・・・私は未だわからないのですが、Wordの場合、出来上がったPDFファイルの順序（ページの順序）が逆である可能性があります。これは、単純な理由で、1page目から順次印刷（=PDF化）して、新しいPDFページは常に上にくるようになっているのですから。

というわけで、それがいやな方は、プリンタの設定の「Acrobat Distiller」のプロパティで、「全般」タブの「印刷設定」→「レイアウト」タブの「ページの順序」を<逆>にする・・・としてみてください。多分、これでOKでしょう。（Acrobat 4.05の時のみと思われる）

その他、ご興味ある方は、[こちら](#)からご一報ください。・・・頑張ってください。