

Exporter vos états Access en PDF gratuitement

par [Charles A.](#)

Date de publication : 01/12/2005

Dernière mise à jour : 01/12/2005

Thèmes abordés : . installation de PDF Creator . installation de PDF Tool Kit
. utilisation VBA de ces deux logiciels . exploitation des documents PDF
Niveau requis : moyen / confirmé

- I - Le PDF gratuit
- II - Installer PDF Creator
 - II-A - Télécharger PDF Creator
 - II-B - Installer PDF Creator
 - II-C - Configurer PDF Creator
- III - Installer PDF Tool Kit
 - III-A - Télécharger PDF Tool Kit
 - III-B - Installer PDF Tool Kit
- IV - Génération manuelle
- V - Génération automatique de PDF
 - V-A - Méthodologie
 - V-B - Changer l'imprimante par défaut
 - V-C - Exporter l'état en PDF
 - V-D - Ajouter l'état à la file d'attente
 - V-E - Renommer un PDF généré automatiquement
 - V-F - Ouverture d'un formulaire par Automation Access
- VI - La gestion avancée des PDF
 - VI-A - Fonctionnement de AccessPDF
 - VI-B - Table de mise en page
 - VI-C - Le sous-formulaire
 - VI-D - Le formulaire
- VII - Une Application portable
- VIII - Conclusion

I - Le PDF gratuit

Pourquoi le PDF ?

C'est une réponse à la question : "comment envoyer un état par mail ?"

Les réponses habituelles sont :

. le format **Snapshot**

. l'export **HTML**

. l'export **Excel**

. l'export **RTF / Word**

Les solutions HTML, Excel, ou même RTF/Word ne sont pas satisfaisantes car elles entraînent une détérioration de la présentation.

Microsoft a bien prévu une solution propriétaire, équivalente au PDF quant à la qualité, mais trop peu de lecteurs de ce format sont installés. Vous pourrez cependant le trouver ici en téléchargement ([SnapShot Viewer](#)).

Comme presque tous les postes sont équipés, à l'heure actuelle d'un lecteur de PDF, la solution la plus pertinente reste donc ce format de fichier.

PDF est l'acronyme de Printable Document Format, un format développé par **Adobe Systems**, dérivé du langage PostScript.

L'idée majeure est de pouvoir échanger des documents à travers n'importe quelle plateforme. Il faut donc pouvoir générer le document et le lire, quels que soient les contextes système et logiciel.

Adobe fournit bien entendu une solution, mais celle-ci est payante.

Ce tutoriel a pour objet de vous présenter une **solution alternative et non payante**.

II - Installer PDF Creator

PDF Creator est selon moi le meilleur logiciel libre de création de documents au format PDF.

Cette section va vous guider pas à pas dans l'installation adéquate du logiciel.

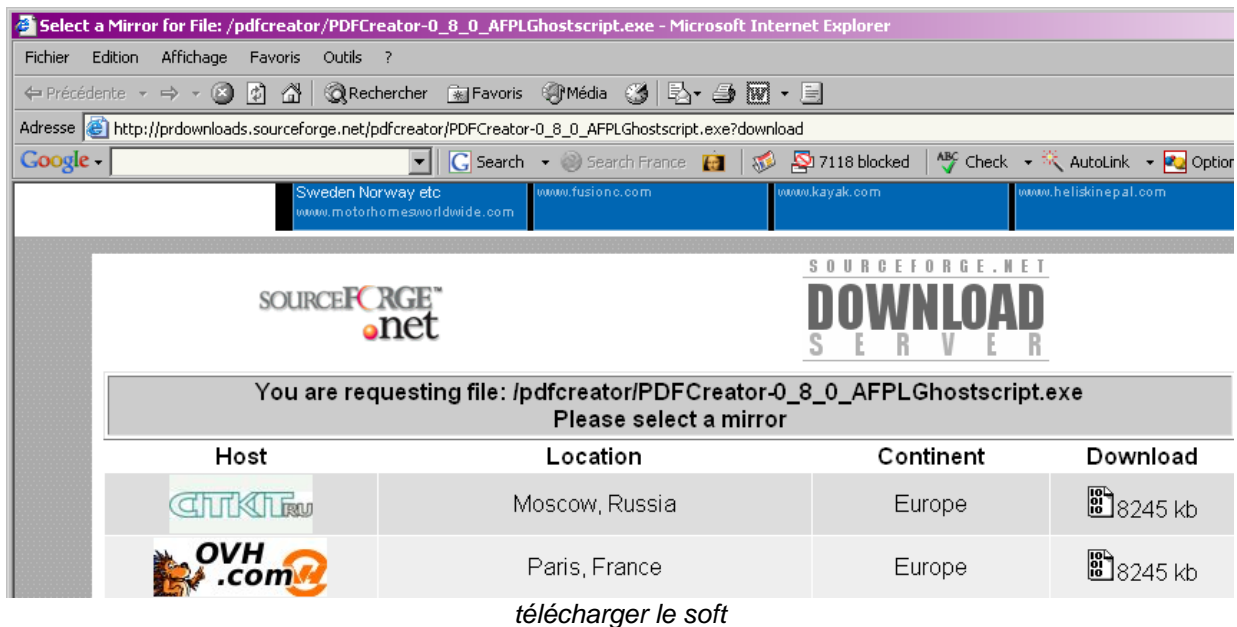
II-A - Télécharger PDF Creator

C'est sur le site de Sector7g, un groupe allemand, que vous allez pouvoir trouver le logiciel.

http://sector7g.wurzel6.de/pdfcreator/index_en.htm

La page de téléchargements est située sur le réseau sourceforge.net : https://sourceforge.net/project/showfiles.php?group_id=57796.

Pour le télécharger directement cliquez : [ici](#).







Select a Mirror for File: /pdfcreator/PDFCreator-0_8_0_AFPLGhostscript.exe - Microsoft Internet Explorer

Adresse http://prdownloads.sourceforge.net/pdfcreator/PDFCreator-0_8_0_AFPLGhostscript.exe?download

SOURCEFORGE.NET
DOWNLOAD SERVER

You are requesting file: /pdfcreator/PDFCreator-0_8_0_AFPLGhostscript.exe
Please select a mirror

Host	Location	Continent	Download
	Moscow, Russia	Europe	 8245 kb
	Paris, France	Europe	 8245 kb

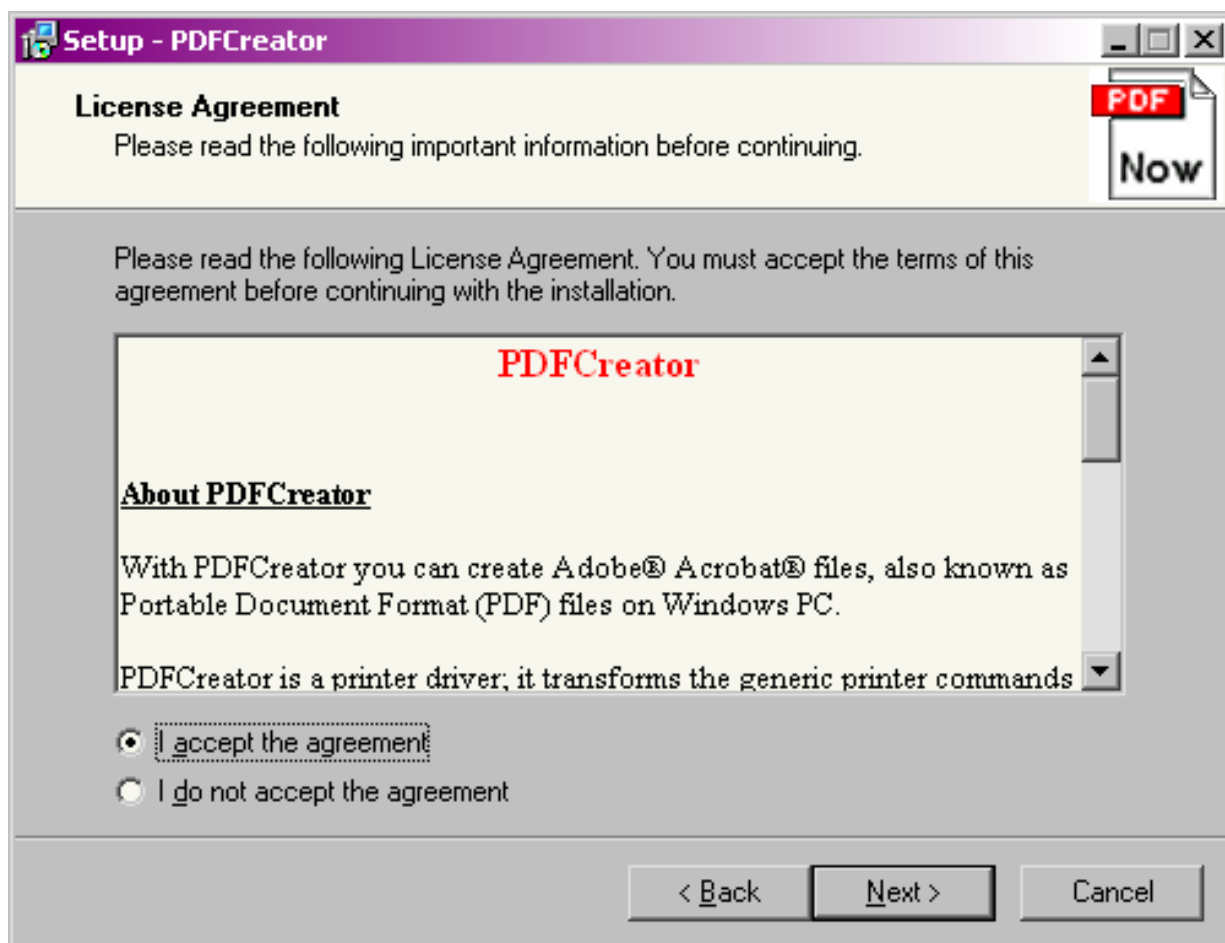
télécharger le soft

Et le lien pour le fichier de langage français : [ici](#).

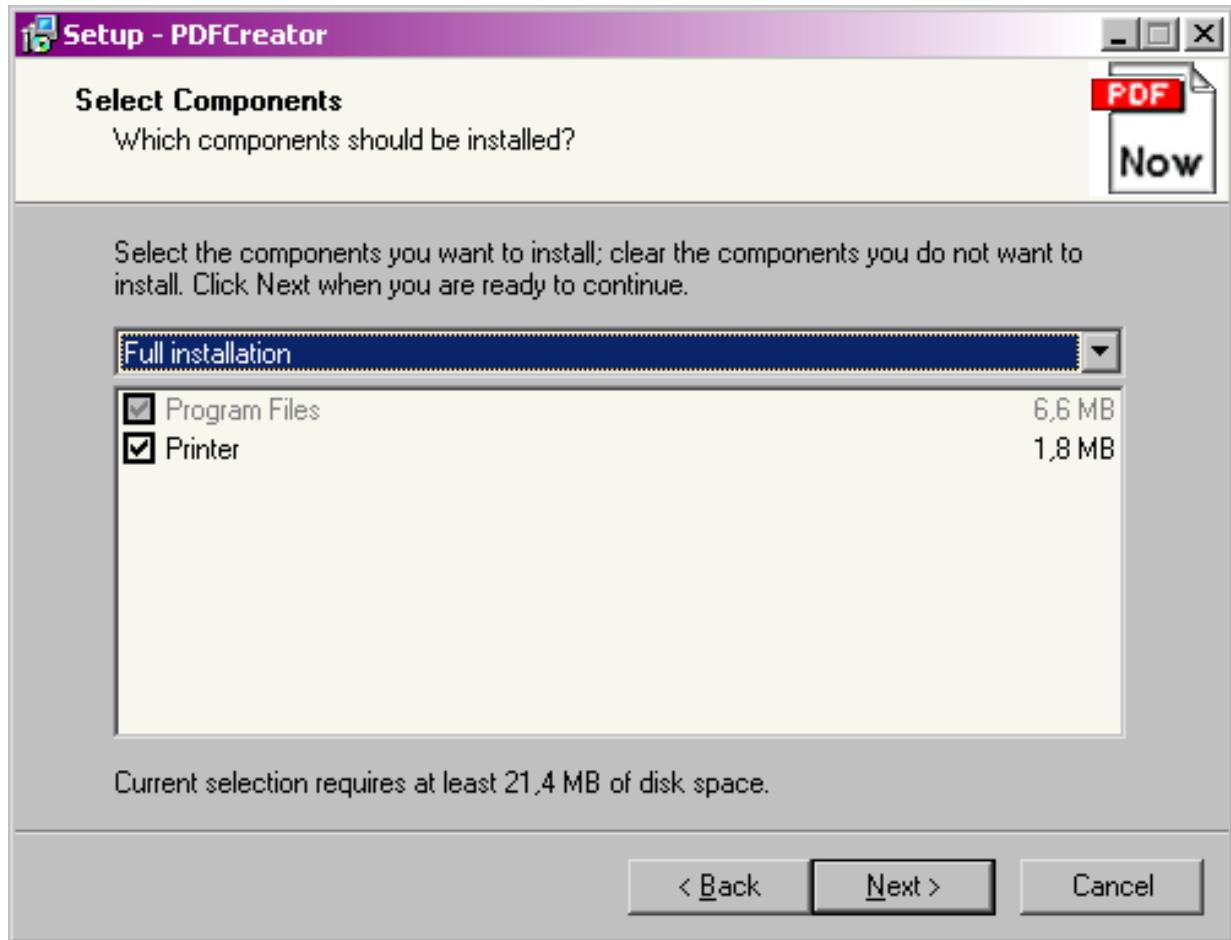
II-B - Installer PDF Creator

Une fois que les téléchargements sont terminés nous passons à l'installation.

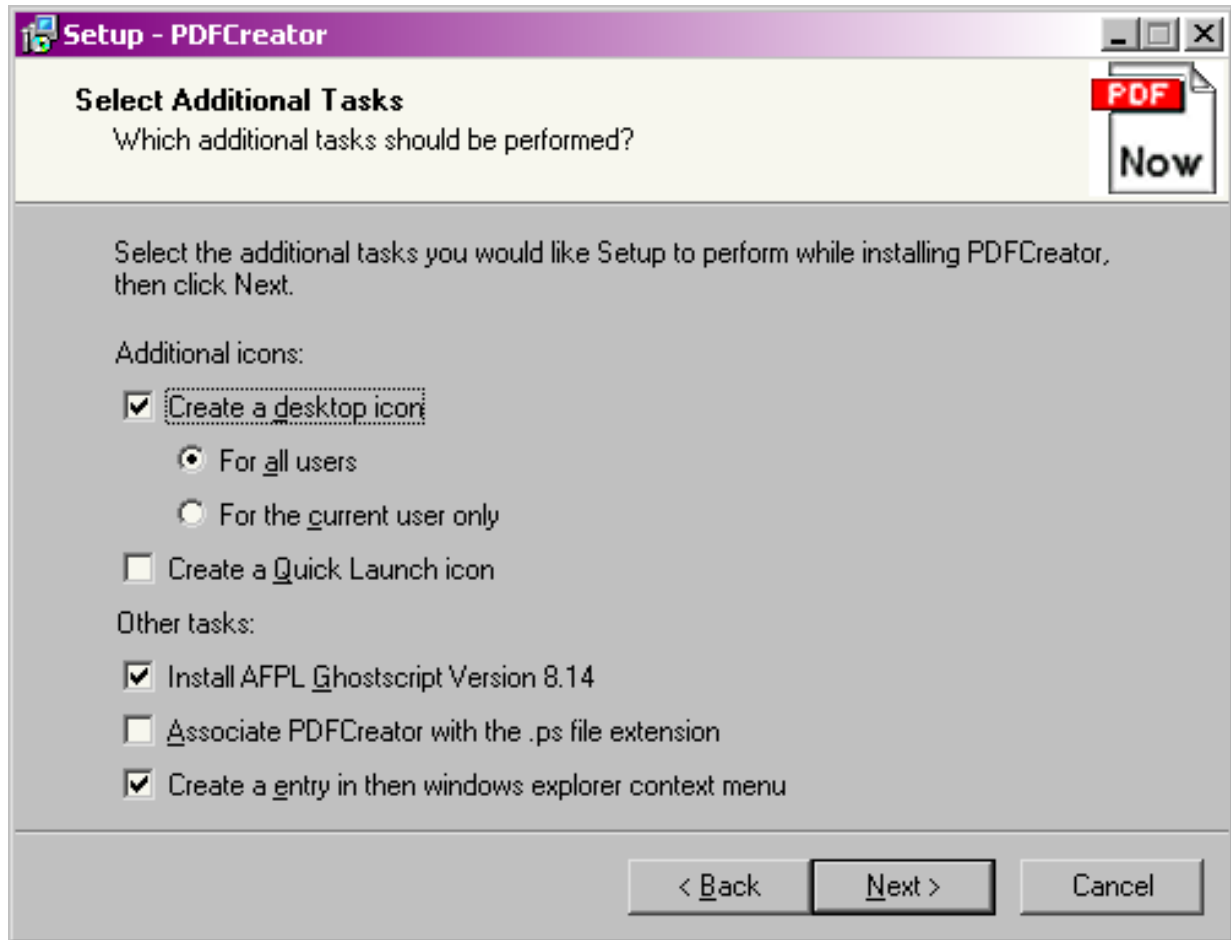
Lançons : **PDFCreator-0_8_0_AFPLGhostscript.exe**



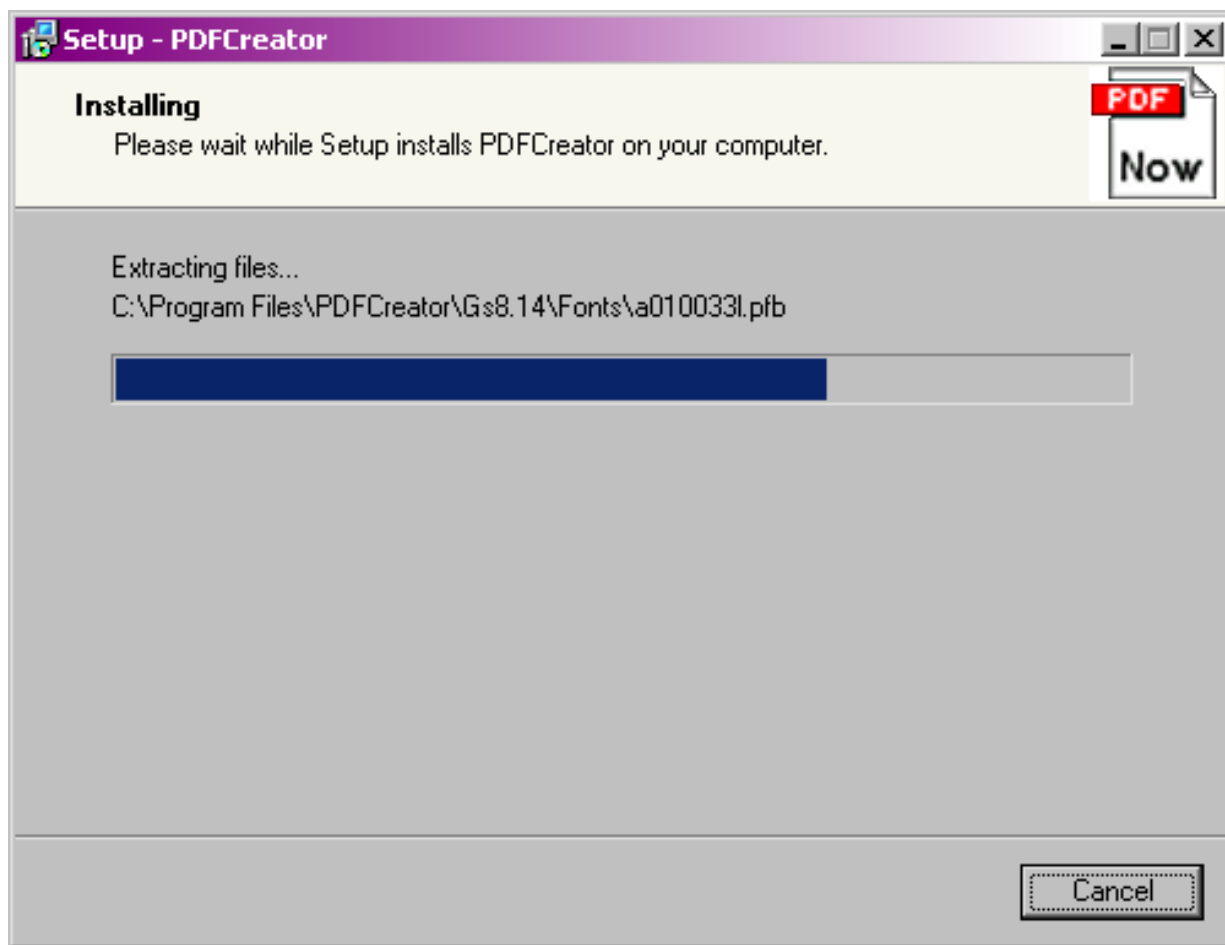
lancer l'installation



choisir les options

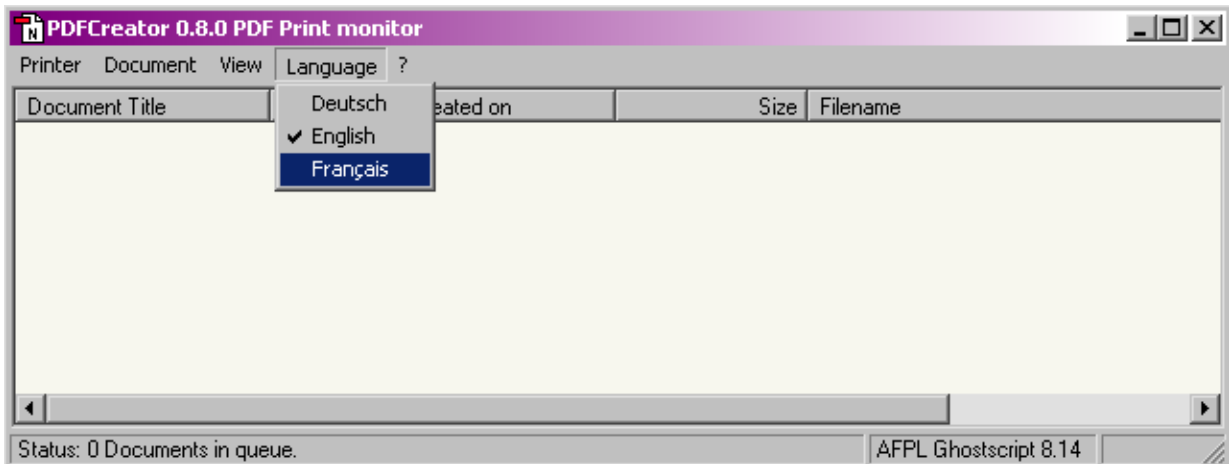


paramétrage



poursuite de l'installation ...

Pour mettre le langage en français, il suffit de copier le fichier **french.ini** dans le répertoire d'installation, ici :
C:\Program Files\PDFCreator\languages

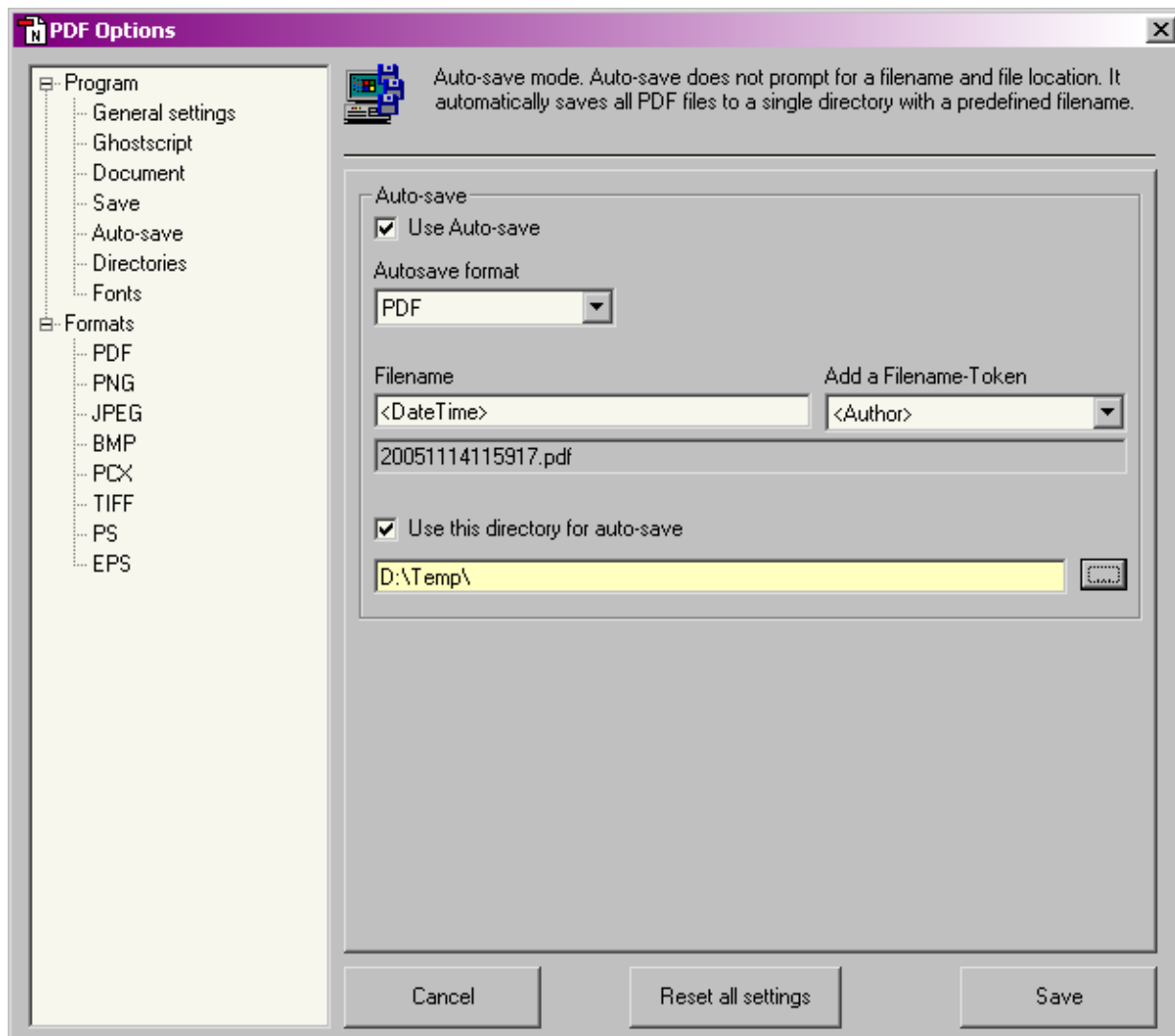


mettre le langage en français

II-C - Configurer PDF Creator

Par le menu : **Printer > Options** nous allons pouvoir configurer correctement PDF Creator.

Nous configurons la rubrique **Auto-save** de la manière suivante



paramétrage auto save

Notre but est d'éviter que le logiciel nous demande où placer systématiquement et par défaut les documents PDF ; pour cela nous définissons ici un répertoire temporaire, et nous spécifions que le nom du fichier sera la concaténation de la date et de l'heure courante.

Nous validons en cliquant sur "**Save**".

III - Installer PDF Tool Kit

Une fois les fichiers PDF créés, nous allons avoir besoin de les fusionner, de les séparer, de les modifier et le logiciel libre le plus simple pour effectuer ces opérations est **PDF Tool Kit**.

III-A - Télécharger PDF Tool Kit

Vous pouvez trouver cet excellent logiciel sur : <http://www.accesspdf.com/pdftk/>

Pour le télécharger directement, cliquez : [ici](#).

III-B - Installer PDF Tool Kit

L'installation est des plus simples, il suffit de décompresser l'archive dans un répertoire précis.

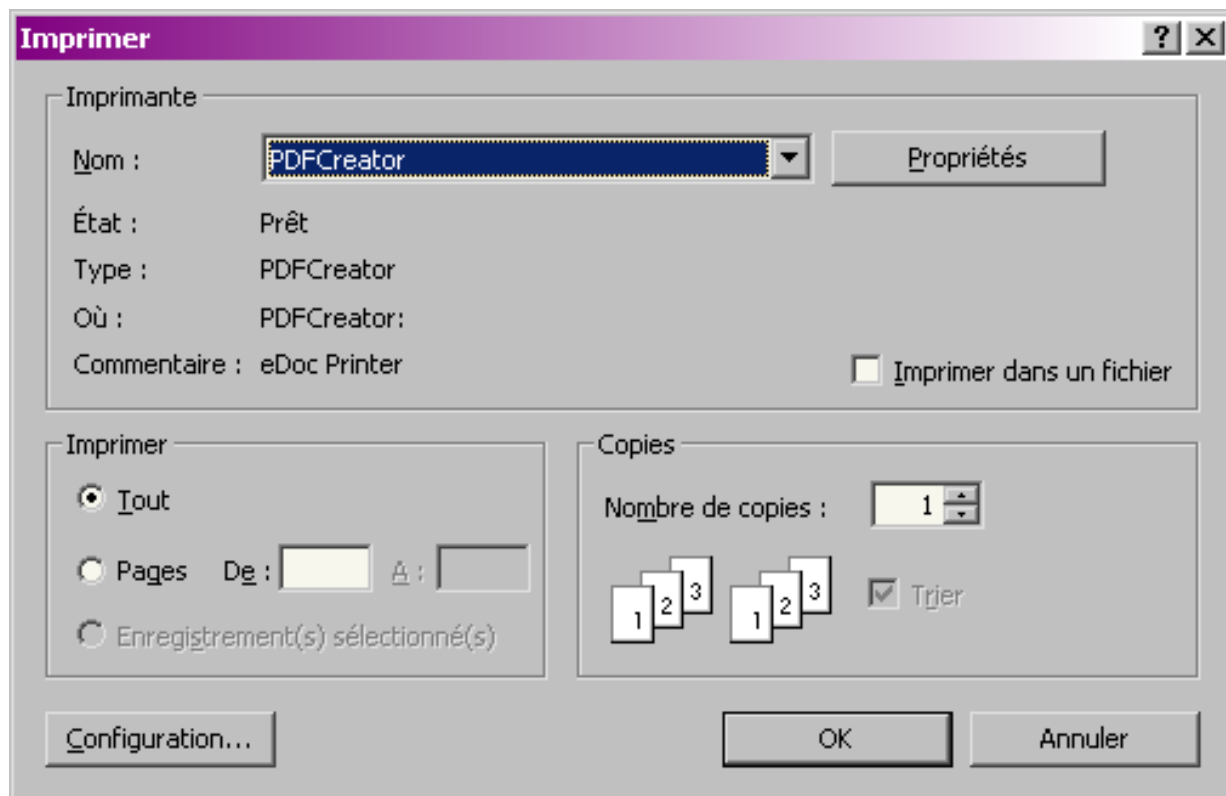
Ici, nous avons choisi : **C:\Program Files\pdfmerge**

Le logiciel est prêt à l'emploi, il ne nécessite pas de paramétrage particulier puisqu'il fonctionne par ligne de commande.

IV - Génération manuelle

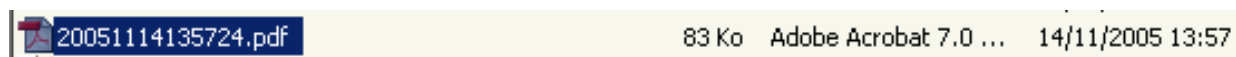
C'est la manière la plus simple de créer un PDF.

Il suffit d'ouvrir un état, puis d'en demander l'impression, choisir l'imprimante virtuelle PDFCreator.



Imprimer

Nous constatons que le fichier a bien été généré à l'endroit prévu ...



Fichier généré

V - Génération automatique de PDF

Nous allons aborder ici le processus qui permet d'**automatiser** la génération de fichiers PDF par programmation pour un **état Access**.

V-A - Méthodologie

Pour créer un document PDF, nous allons procéder par étape :

- . récupération de l'imprimante courante ;
- . définition de l'imprimante par défaut sur PDFCreator ;
- . impression des pages désirées ;
- . rétablissement de l'ancienne imprimante ;
- . renommage du document créé.

Nous avons défini dans les options, de nommer le fichier de sortie en fonction de l'heure courante et non en fonction du nom du document.

Pourquoi ? Parce que nous pouvons très bien lancer plusieurs traitements à la suite et que certains peuvent prendre beaucoup plus de temps que d'autres.

Nous allons donc stocker dans une table, la file d'attente des créations de PDF à générer.

V-B - Changer l'imprimante par défaut

La création du format **PDF**, dérivé du **PostScript**, se fait par l'intervention d'une imprimante virtuelle.

Le but de cette portion de code est de modifier les paramètres d'impression définis dans Windows.

Pour changer ces paramètres, il est nécessaire d'écrire dans le fichier **win.ini** afin d'y déclarer la nouvelle imprimante.

changement d'imprimante par défaut

```

Option Explicit
Const HWND_BROADCAST = &HFFFF
Const WM_WININICHANGE = &H1A

'
'
' API kernel32.dll
'
' lire dans un fichier INI
Private Declare Function apiGetPrivateProfileString Lib "kernel32" _
    Alias "GetPrivateProfileStringA" ( _
        ByVal lpApplicationName As String, _
        ByVal lpKeyName As Any, ByVal lpDefault As String, _
        ByVal lpReturnedString As String, ByVal nSize As Long, _
        ByVal lpFileName As String) As Long

' renvoyer le répertoire Windows
Private Declare Function apiGetWindowsDirectory Lib "kernel32" _
    Alias "GetWindowsDirectoryA" ( _
        ByVal lpBuffer As String, ByVal nSize As Long) As Long

' écrire dans un fichier INI
Private Declare Function apiWritePrivateProfileString Lib "kernel32" _
    Alias "WritePrivateProfileStringA" ( _
        ByVal lpApplicationName As String, _
        ByVal lpKeyName As Any, ByVal lpString As Any, _
        ByVal lpFileName As String) As Long

'
' API user32.dll
'
' envoyer un message qui indique à Windows la mise à jour de WIN.INI
Private Declare Function apiSendMessage Lib "user32" _
    Alias "SendMessageA" ( _
        ByVal hwnd As Long, ByVal wParam As Long, _
        ByVal lParam As Integer, ByVal lParam As Any) As Long

Private strPath As String
Private lngNC As Long
Private strRet As String

Sub SwitchDefaultPrinter(Nom As String)
    ' modifie le nom de l'imprimante par défaut

    strPath = String(260, 0)
    ' récupère le chemin de win.ini
    strPath = Left$(strPath, apiGetWindowsDirectory(strPath, Len(strPath))) + "\win.ini"
    strRet = String(255, 0)
    lngNC = apiGetPrivateProfileString("Devices", Nom, "", strRet, 255, strPath)
    strRet = Left(strRet, lngNC)
    ' écrit dans win.ini le nom de l'imprimante souhaitée
    apiWritePrivateProfileString "windows", "device", Nom & ",", strRet, strPath
    ' signale à MS Windows de prendre en compte la modification de win.ini
    apiSendMessage HWND_BROADCAST, WM_WININICHANGE, 0, "windows"

End Sub

Function GetDefaultPrinter() As String
    ' renvoie le nom de l'imprimante par défaut

    strPath = String(260, 0)
    strPath = Left$(strPath, apiGetWindowsDirectory(strPath, Len(strPath))) + "\win.ini"
    strRet = String(255, 0)
    lngNC = apiGetPrivateProfileString("windows", "device", "", strRet, 255, strPath)
    strRet = Left(strRet, lngNC)
    lngNC = InStr(strRet, ",")
    GetDefaultPrinter = Left(strRet, lngNC - 1)

End Function

```

V-C - Exporter l'état en PDF

Obtenir le PDF d'un état

```

Function getPDF(ByVal strReport As String, _
                Optional ByVal allPages As Integer = acPrintAll, _
                Optional ByVal pStart As Integer, _
                Optional ByVal pEnd As Integer)

Dim strOldPrinter As String
Dim strPdfPrinter As String
Dim dblStamp As Double

' récupération de l'imprimante par défaut
strOldPrinter = GetDefaultPrinter

' attribution de l'imprimante PDF
strPdfPrinter = "PDFCreator"
SwitchDefaultPrinter strPdfPrinter

' récupération de la date / heure courante
dblStamp = Now
' ajout du document dans la file d'attente
addDocument strReport & IIf(allPages <> acPrintAll, " [" & pStart & "-" & pEnd & "]", ""), dblStamp

' ouverture de l'état par automation
fOpenRemoteReport workDB.Name, strReport, _
                  acViewPreview, allPages, _
                  IIf(allPages <> acPrintAll, pStart, 1), IIf(allPages <> acPrintAll, pEnd, 9999)

' réattribution de l'imprimante par défaut
SwitchDefaultPrinter strOldPrinter

' gestion de la file d'attente des documents
ScanPDFfiles

End Function

```

La fonction contient les arguments suivants :

. *strReport* (**Nom de l'état**) : nom de l'état à exporter en PDF ;

. *allPages* (**Type de page**) (*acPrintAll* par défaut) : détermine si nous imprimons toutes les pages ou simplement une section ;

. *pStart* (**page de départ**) et *pEnd* (**page de fin**) : dans le cas où nous n'imprimons pas toutes les pages

Nous ouvrons l'état grâce à la commande **fOpenRemoteReport** qui prend en charge les opérations d'ouverture / d'impression / de fermeture de l'état.

Nous allons détailler dans les paragraphes suivants les fonctions auxiliaires suivantes :

. **AddDocument** nous permet d'ajouter notre impression dans la table de file d'attente des PDF.

. **ScanPDFfiles** dont le but est de vider la file d'attente des documents PDF.

. **fOpenRemoteReport** qui sert à ouvrir un état d'une autre base de données par **Automation Access**.

V-D - Ajouter l'état à la file d'attente

Ajout d'un état à la file d'attente PDF

```
Function addDocument(ByVal strDocName As String, ByVal dblStamp As Double)
    ' ajouter un document dans la file d'attente

    If Not isTable("tblPDFdoc") Then
        ' création de table
        DoCmd.RunSQL "CREATE TABLE tblPDFdoc (doc TEXT, tim TEXT, done YESNO);"
    End If

    ' SQL d'insertion
    DoCmd.RunSQL "INSERT INTO tblPDFdoc VALUES (" & strDocName & ", " & Format(dblStamp,
"yyyyymmddhhnnss") & ", 0);"

End Function

Function isTable(tblName As String) As Boolean
    ' tester l'existence d'une table

    On Error GoTo istblerr

    Debug.Print CurrentDb.TableDefs(tblName).Name
    isTable = True
    Exit Function

    istblerr:
    isTable = False
    Err.Clear

End Function
```

Nous allons alors ajouter dans la table **tblPDFdoc** le nom de l'état, l'heure ainsi que la date courante et ce afin de le rapprocher du nom du PDF généré automatiquement par PDFCreator.

Le champ **[done]** nous passons la valeur à False et prend la valeur **True** dès que l'état a été généré en PDF.

V-E - Renommer un PDF généré automatiquement

Gestion de la file d'attente des PDF

```
Sub ScanPDFfiles()
    ' traitement des fichiers en file d'attente

    Dim strPath As String, currFile As String
    Dim rec As DAO.Recordset
    Dim fso As New FileSystemObject
    Dim intCount As Integer

    On Error GoTo scanPDF

    intCount = 0
    ' nous avons stocké le chemin de une propriété de la base
    strPath = CurrentDb.Properties("workPath")
    Set rec = CurrentDb.OpenRecordset("SELECT * FROM tblPDFdoc WHERE done = False ORDER BY tim;",
dbOpenDynaset)
    Do While Not rec.EOF
```

Gestion de la file d'attente des PDF

```

' fonction pour trouver le fichier dont la date est la plus proche
' de la date de demande d'édition
currFile = GetFirstFileName(rec!tim)
If Len(currFile) > 0 Then
    ' si le fichier a été trouvé on le renomme
    fso.MoveFile strPath & currFile, strPath & rec!doc & IIf(intCount = 0, "", intCount) &
    ".pdf"
    intCount = 0
    ' mise à jour de la table de la file d'attente
    rec.Edit
    rec!done = True
    rec.Update
End If
rec.MoveNext
Loop

rec.Close
Set rec = Nothing
Set fso = Nothing

Exit Sub

' traitement d'erreurs
scanPDF:
If Err.Number = 58 Then
    ' si le fichier existe déjà
    ' on rajoute un numéro au nom ...
    intCount = intCount + 1
    Resume
Else
    MsgBox Err.Number & " - " & Err.Description
    Err.Clear
    Resume Next
End If

Set fso = Nothing

End Sub

Function GetFirstFileName(ByVal strStamp As String) As String
' fonction renvoyant le nom du fichier dont la date est la plus proche
' de la date de demande d'édition

Dim strPath As String
Dim strFic As String
Dim dblTargetFic As Double

' valeur maximum
dblTargetFic = CDBl("29991231235959")

' nous avons stocké le chemin de une propriété de la base
strPath = CurrentDb.Properties("workPath")

strFic = Dir(strPath & Left(strStamp, 4) & "*.pdf")
Do While Len(strFic) > 0
    If CDBl(Left(strFic, Len(strFic) - 4)) >= CDBl(strStamp) Then
        ' parmi les fichiers dont la date est supérieure ou égale
        ' à la date de demande d'édition, on prend celle qui a la valeur
        ' minimale
        dblTargetFic = Minus(dblTargetFic, CDBl(Left(strFic, Len(strFic) - 4)))
    End If
    strFic = Dir
Loop

GetFirstFileName = IIf(dblTargetFic <> 29991231235959#, dblTargetFic & ".pdf", "")

End Function

Function Minus(vA, vB)
' trouver le minimum entre deux valeurs

If vA > vB Then
    Minus = vB
Else
    Minus = vA
End If

End Function

```

Dans les options que nous avons définies sur **PDFCreator**, le fichier prend automatiquement le nom défini avec la date et l'heure courante suivi de l'extension PDF.

Par exemple, lors d'une génération de fichier à 23h08m54s le 14/11/2005, le fichier s'appellera **20051114230854.pdf**

Notre code permet de chercher le fichier portant le nom dont la date est la plus proche de la date de demande de génération tout en étant supérieure.

Une fois ce fichier trouvé, grâce à la fonction **GetFirstFileName()**, nous allons changer le nom du fichier.

La modification du nom est effectuée par code en utilisant les fonctionnalités de l'objet **FileSystemObject** issu de la DLL *Microsoft Scripting Runtime*.

Pour renommer les fichiers, nous employons la méthode **.MoveFile** .

Dès que le fichier est renommé, nous modifions le statut du document dans la file d'attente en passant le champ **Done** à *True*.

V-F - Ouverture d'un formulaire par Automation Access

Nous avons besoin pour notre application d'ouvrir un état situé sur une autre base que celle en cours.

Nous allons utiliser pour cela l'**automation Access**.

Le fonctionnement est le même pour l'automation Excel plus connue, hormis que cette fois, il ne sera pas nécessaire de rajouter la référence aux objets Access puisque nous l'utilisons par défaut.

Une fois l'objet de base Access instancié, nous allons ouvrir, imprimer puis fermer l'état avant de quitter la base.

Il ne faudra pas oublier de libérer de la mémoire les objets que nous avons utilisés.

Ouverture et impression d'un état par Automation

```
Function fOpenRemoteReport(ByVal strMDB As String, _
    ByVal strReport As String, _
    ByVal aMode, ByVal aPage, _
    ByVal iStart As Integer, ByVal iEnd As Integer) As Boolean
    Dim objAccess As Access.Application
    Dim lngRet As Long

    ' gestion d'erreurs
```

Ouverture et impression d'un état par Automation

```

On Error GoTo fOpenRemoteReport_Err

If Len(Dir(strMDB)) > 0 Then
    ' creation de l'objet Access
    Set objAccess = New Access.Application
    With objAccess
        'ouverture de la base
        .OpenCurrentDatabase strMDB
        'les commandes sont les memes que pour la base en cours
        ' hormis le "objAccess."
        ' ouverture de l'état
        .DoCmd.OpenReport strReport, aMode
        ' impression des pages
        .DoCmd.PrintOut aPage, iStart, iEnd, acHigh
        ' fermeture de l'état sans sauvegarde
        .DoCmd.Close acReport, strReport, acSaveNo
    End With
End If

fOpenRemoteReport_Exit:
    ' libération des objets
    On Error Resume Next
    objAccess.Quit
    Set objAccess = Nothing
    Exit Function

fOpenRemoteReport_Err:
    fOpenRemoteReport = False
    Select Case Err.Number
        Case 7866:
            'mdb ouverte en mode exclusif
            MsgBox "The database you specified " & vbCrLf & strMDB & _
                vbCrLf & "is currently open in exclusive mode. " & vbCrLf & _
                & vbCrLf & "Please reopen in shared mode and try again", _
                vbExclamation + vbOKOnly, "Could not open database."
        Case 2103:
            'l'état n'existe pas
            MsgBox "The report '" & strReport & _
                "' doesn't exist in the Database " _
                & vbCrLf & strMDB, _
                vbExclamation + vbOKOnly, "report not found"
        Case 7952:
            'l'utilisateur a fermé le fichier mdb
            fOpenRemoteReport = True
        Case Else:
            MsgBox "Error#: " & Err.Number & vbCrLf & Err.Description, _
                vbCritical + vbOKOnly, "Runtime error"
    End Select
    Resume fOpenRemoteReport_Exit
End Function

```

VI - La gestion avancée des PDF

Il arrive que la demande ne se limite pas au simple export d'un état.

Les problèmes rencontrés suite à l'envoi de PDF bruts :

- . pages inutiles ;
- . nécessité d'envoyer plusieurs documents à la suite ;
- . répétition de certaines pages (type page de garde).

Ces limitations font apparaître le besoin de gérer la mise en page des PDF en les découpant et en les fusionnant.

Nous avons installé pour cela l'excellent outil **PDF Tool Kit**.

VI-A - Fonctionnement de AccessPDF

PDFTK de AccessPDF est un logiciel de manipulation de PDF par ligne de commande.

Le script comprend trois parties :

- . l'exécutable suivi des différents noms de fichiers, chacun représentés par une lettre,
- . l'ordre de mise en page précédé du mot clé **cat**, dans l'exemple le document B, puis le A, puis le C sur la section allant de page 1 à page 2.
- . le nom du fichier de sortie précédé du mot clé **output**.

Fonctionnement par l'exemple :

Script Batch de gestion de PDF

```
"C:\Program Files\pdfmerge\pdftk.exe" A="D:\Temp\toto.pdf" B="D:\Temp\tata.pdf" C="D:\Temp\tutu.pdf"  
cat B A C1-2  
output "D:\Temp\Export.pdf"
```

VI-B - Table de mise en page

Pour gérer efficacement les PDF nous allons utiliser une table et un formulaire ainsi qu'un sous-formulaire.

SQL de création de table

```
CREATE TABLE tblPDFwork (id COUNTER, dnom TEXT, pfrom INTEGER, pto INTEGER, docorder INTEGER);
```

id est un numéro autoincrémenté.

dnom est le nom du document

pfrom est la page de départ (-1 si nous avons besoin de toutes les pages)

pto est la page de fin (-1 si nous avons besoin de toutes les pages)

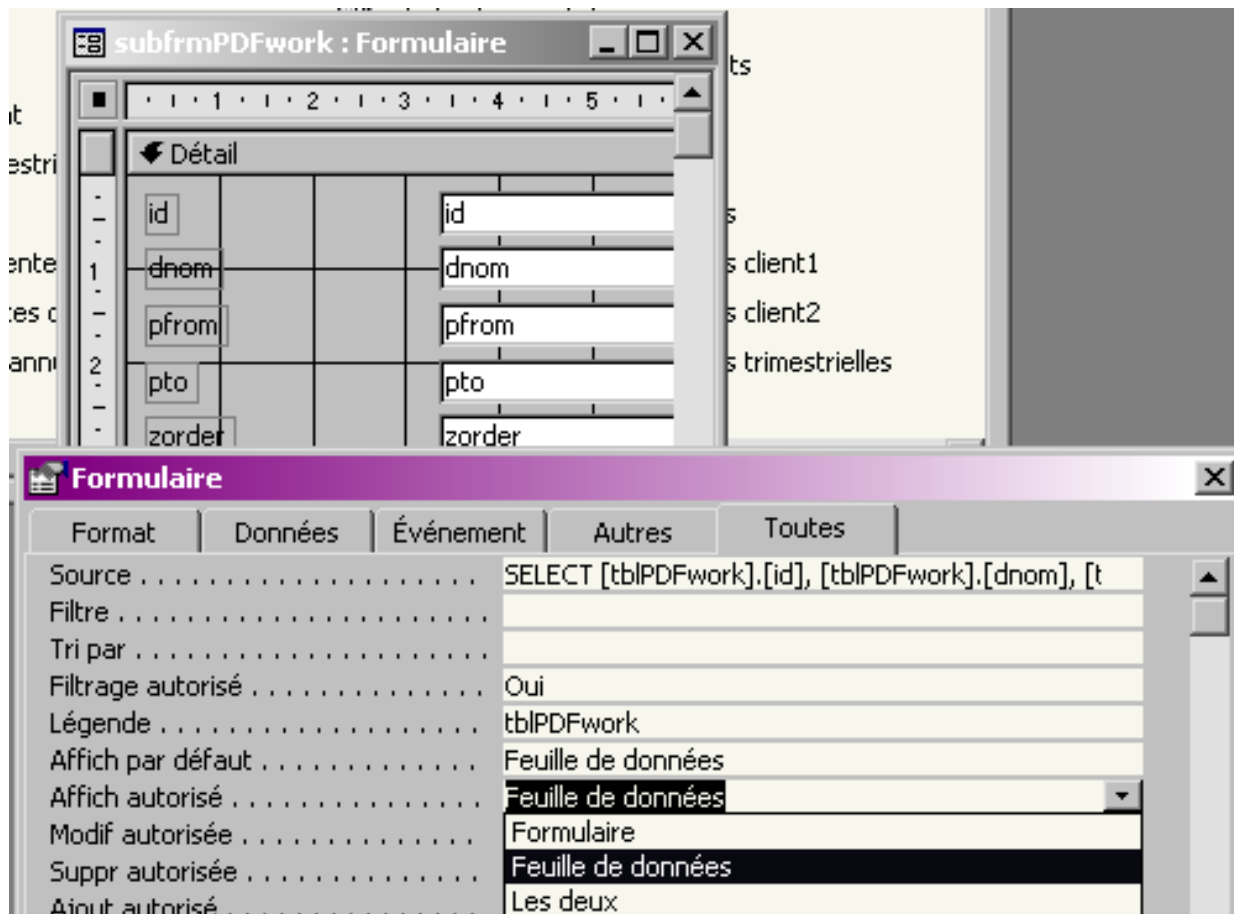
docorder est un numérique servant à donner l'ordre de passage des documents.

VI-C - Le sous-formulaire

Nous créons par l'assistant un formulaire instantané à partir de la table que nous venons de créer **tblPDFwork**.

Nous nommons le formulaire : **subfrmPDFwork**.

Nous forçons l'affichage en mode feuille de données, ce sera notre sous-formulaire de saisie des membres du PDF compilé.



forçage du mode feuille de données

La source de données est le SQL suivant, un simple tri sur le champ **zorder** :

SQL du sous-formulaire

```
SELECT tblPDFwork.id, tblPDFwork.dnom, tblPDFwork.pfrom, tblPDFwork.pto, tblPDFwork.zorder
FROM tblPDFwork
ORDER BY tblPDFwork.zorder;
```

En mode feuille de données nous réduisons à zéro la première colonne qui affiche l'Id, de manière à obtenir ceci.

	dnom	pfron	pto	zorder
▶	Catalogue.pdf	-1	-1	1
	Facture.pdf	1	50	2
*				

sous-formulaire ajustement des colonnes

VI-D - Le formulaire

Nous créons un formulaire indépendant comprenant :

- . le sous-formulaire **subfrmPDFwork** ;
- . une liste des fichiers PDF sur le disque : **lstFiles** ;
- . des boutons ajout / suppression (**cmdAdd**, **cmdDel**) pour ajouter ou enlever un pdf de la compilation ;
- . deux textbox (**txtFrom**, **txtTo**) pour saisir les pages concernées ;
- . une textbox pour indiquer le nom de la compilation (**txtProjectName**) ;
- . un bouton pour lancer le script de génération : **cmdGenerate**.

Gestion du chargement du formulaire

```
Private Sub Form_Load()
' dans le cas où la table n'existe pas, nous la créons.
If Not isTable("tblPDFwork") Then
DoCmd.RunSQL "CREATE TABLE tblPDFwork (id COUNTER, dnom TEXT, pfrom INTEGER, pto INTEGER, zorder
INTEGER);"
End If

' remplissage de la liste
FillList

End Sub

Sub FillList()
' remplissage de la liste des fichiers PDF dans notre répertoire de travail
Dim strPath As String
Dim strFic As String
Dim strSRC As String

strPath = "D:\Temp\"
strFic = Dir(strPath & "*.pdf")
Do While Len(strFic) > 0
strSRC = strSRC & ";" & strFic
strFic = Dir
Loop

Me.lstFiles.RowSource = Right(strSRC, Len(strSRC) - 1)

End Sub

Private Sub subfrmPDFwork_Exit(Cancel As Integer)
' lorsque le sous-form perd le focus, nous remettons à jour
' utile en particulier lorsque l'utilisateur change l'ordre.
Me.subfrmPDFwork.Form.Requery
End Sub
```


Gestion du chargement du formulaire

```
End Sub
```

Le code suivant permet de gérer les mises à jour (ajout / suppression) de la liste de la compilation de PDF.

Pour info : remettre à jour un sous-formulaire : **Me.NomduSousForm.Form.Requery**

Gestion des boutons ajout et suppression

```
Private Sub cmdAdd_Click()
' ajout d'une ligne par SQL INSERT
If Len(Me.lstFiles) > 0 Then
DoCmd.RunSQL "INSERT INTO tblPDFwork ( dnom , pfrom , pto ) " & _
"VALUES (" & Me.lstFiles & ", " & _
IIf(Len(Me.txtFrom) > 0, Me.txtFrom, -1) & ", " & _
IIf(Len(Me.txtTo) > 0, Me.txtTo, -1) & ");"
' mise à jour du sous-formulaire de la liste de compilation
Me.subfrmPDFwork.Form.Requery
End If

End Sub

Private Sub cmdDel_Click()
' suppression de ligne sélectionnée
If Len(Me.subfrmPDFwork.Form.Id) > 0 Then
DoCmd.RunSQL "DELETE * FROM tblPDFwork WHERE id = " & Me.subfrmPDFwork.Form.Id & ";";
' mise à jour du sous-formulaire de la liste de compilation
Me.subfrmPDFwork.Form.Requery
End If

End Sub
```

Le code suivant crée le PDF à partir des trois parties nécessaires au script de AccessPDF.

Génération de compilation de PDF

```
Private Sub cmdGenerate_Click()

' déclarations
Dim rec As DAO.Recordset
Dim iCount As Integer
Dim strCommand As String
Dim strPath As String
Dim strCat As String
Dim strOutput As String
Dim strScript As String

' initialisations des variables
strPath = "D:\Temp\"
strCommand = "\"C:\Program Files\pdfmerge\pdftk.exe" "
strCat = " cat "
strOutput = " output \"" & strPath & Me.txtProjectName & ".pdf\""
iCount = 65

' ouverture du recordset sur la table de travail, tri en fonction du champ zorder
Set rec = CurrentDb.OpenRecordset("SELECT * FROM tblPDFwork ORDER BY zorder;", dbOpenSnapshot)

Do While Not rec.EOF
' en incrémentant iCount nous permettons d'afficher A, B, C etc.
strCommand = strCommand & Chr(iCount) & "=" & strPath & rec!dnom & " "
' dans le cas où nous restreignons des pages
' nous accolons à la lettre la page de debut - page de fin
```

Génération de compilation de PDF

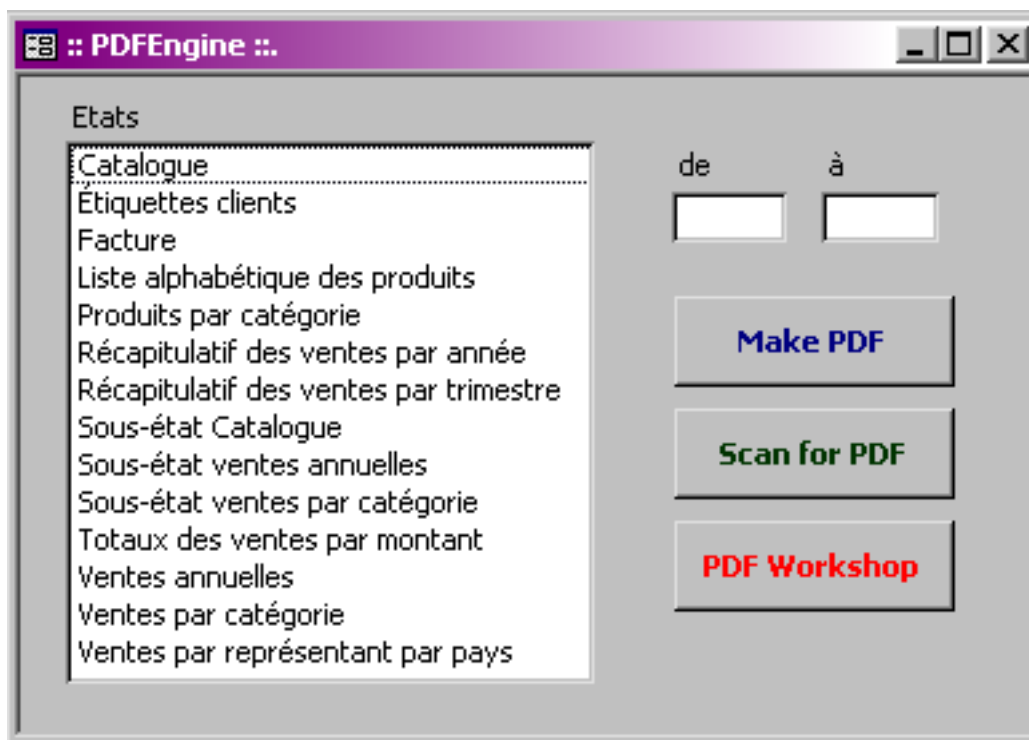
```
    strCat = strCat & Chr(iCount) & IIf(rec!pfrom <> -1, rec!pfrom & "-" & rec!pto, "") & " "  
    iCount = iCount + 1  
    rec.MoveNext  
Loop  
rec.Close  
  
' concaténation des trois parties du script  
strScript = strCommand & strCat & strOutput  
  
' exécution  
Shell strScript, vbHide  
  
Set rec = Nothing  
  
' remise à jour de la liste des pdf  
FillList  
  
End Sub
```

VII - Une Application portable

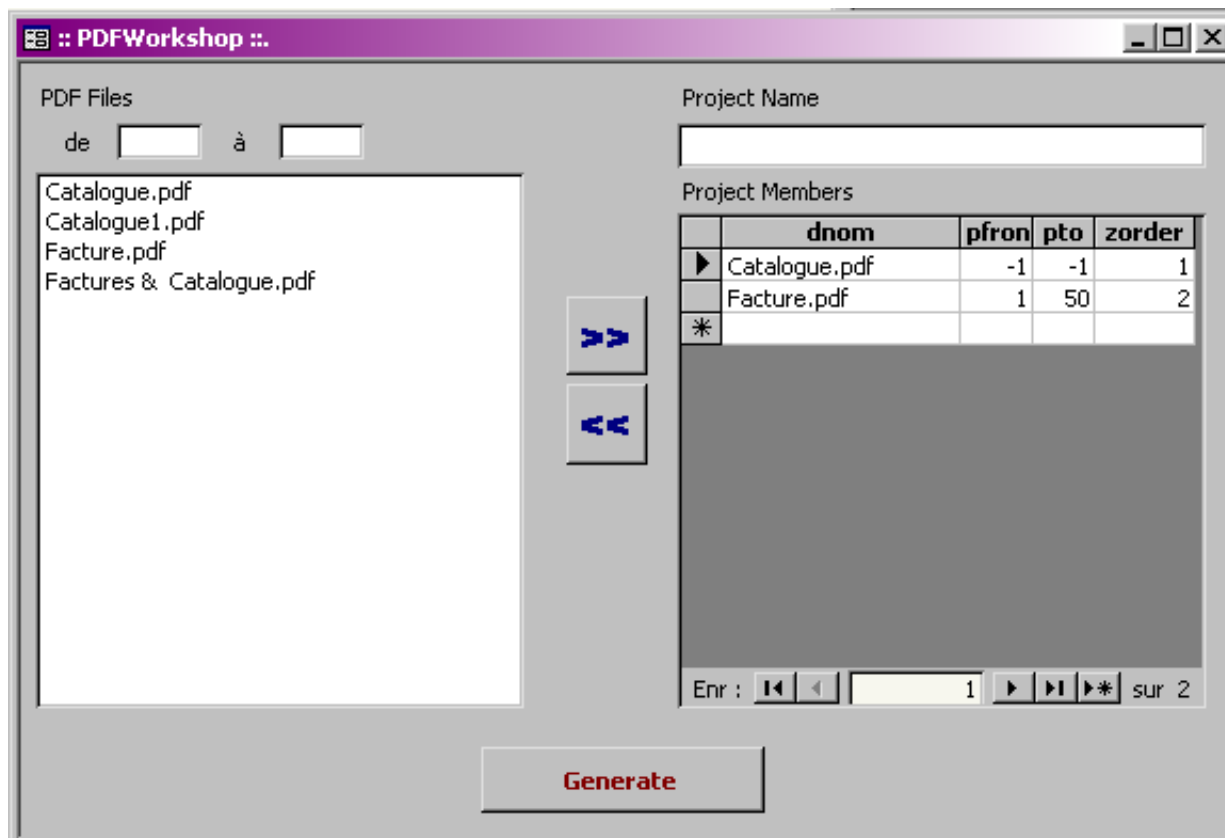
L'application AccessPDF assure de manière autonome la production de documents PDF mais elle peut tout aussi bien être intégrée à une base existante.

Dans le cas d'une intégration, voici les différents éléments à ajouter :

- . **adapter** le code de la **fonction GetPDF** pour ouvrir un état local et non un état d'une base distante ;
- . table **tblPDFdoc** : qui contient la file d'attente des demandes de génération de PDF ;
- . table **tblPDFwork** : qui contient la liste des fichiers d'une compilation de PDF ;
- . formulaire **frmPDFengine** qui permet d'exporter un état en PDF ;
- . formulaire **frmPDFworkshop** qui permet de créer ou éditer une compilation de PDF ;
- . sous-formulaire **subfrmPDFwork** qui liste les membres d'une compilation ;
- . les différents modules **modPDF**, **modDialogBox**, **modSettings**, **modOpenReport** qui contiennent les routines de création et de gestion.



le formulaire frmPDFEngine



le formulaire frmPDFWorkshop

VIII - Conclusion

Vous pouvez télécharger l'application **AccessPDF** en cliquant [ici](#) (version 2000 - 72 ko).

Cet article a pour but de montrer que le format PDF est un excellent moyen d'échanger de l'information et de pallier à une des grandes lacunes d'Access, la distribution d'états.

Ce format est désormais accessible à tous grâce au logiciel libre.

L'atelier PDF que nous avons mis au point pourra facilement être transposé et adapté, on peut aisément imaginer l'ajout d'une fonctionnalité d'envoi automatique de mail.

Je remercie les développeurs d'outils libres de qualité comme les équipes de **Sector7g** (développeurs de PDFCreator) et **Sid Steward** (l'auteur de PDF Tool Kit).

Je remercie également l'équipe de la rédaction Access de developpez.com pour ses relectures et ses pertinentes remarques.