

Enregistrez vos états au format PDF avec Office 2007



par [Christophe WARIN](#)

Date de publication : 18/12/2006

Dernière mise à jour :

La nouveauté tant attendue de Microsoft Access 2007 ! Même si le format PDF n'est pas pris en charge nativement, rassurez-vous, Microsoft propose un complément pour exporter vos documents Office avec une grande facilité.

- I - Introduction
- II - Vérification et installation de l'extension PDF
 - II-A - Vérification
 - II-B - Installation de l'extension
- III - Enregistrer un état au format PDF
 - III-A - Manuellement
 - III-B - En VBA
- IV - Créer un bouton dans le ruban
- V - Conclusion

I - Introduction

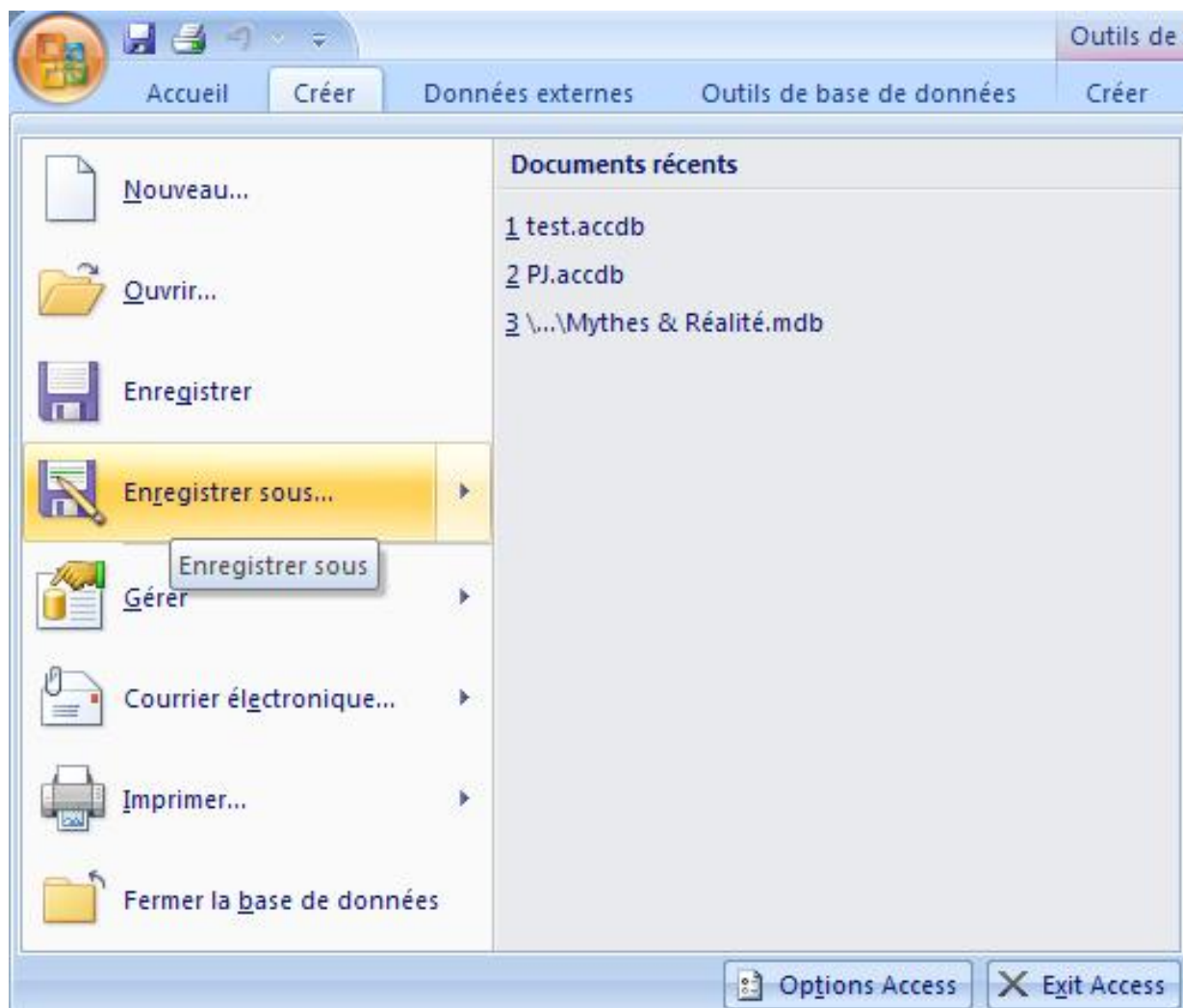
L'export **PDF** des états a été un des premiers besoins exprimés par les utilisateurs de Microsoft Access pour cette nouvelle version. En effet à l'heure actuelle et avec les anciennes versions (par exemple 2003), il est impossible de partager un document avec un autre utilisateur en utilisant un format **portable**. Certains développeurs ont alors mis en place des solutions **complexes** pour pallier ce manque et permettre d'exporter les états en PDF.

Microsoft a pris en compte la demande des utilisateurs et propose d'enregistrer les états sous ce format.

II - Vérification et installation de l'extension PDF

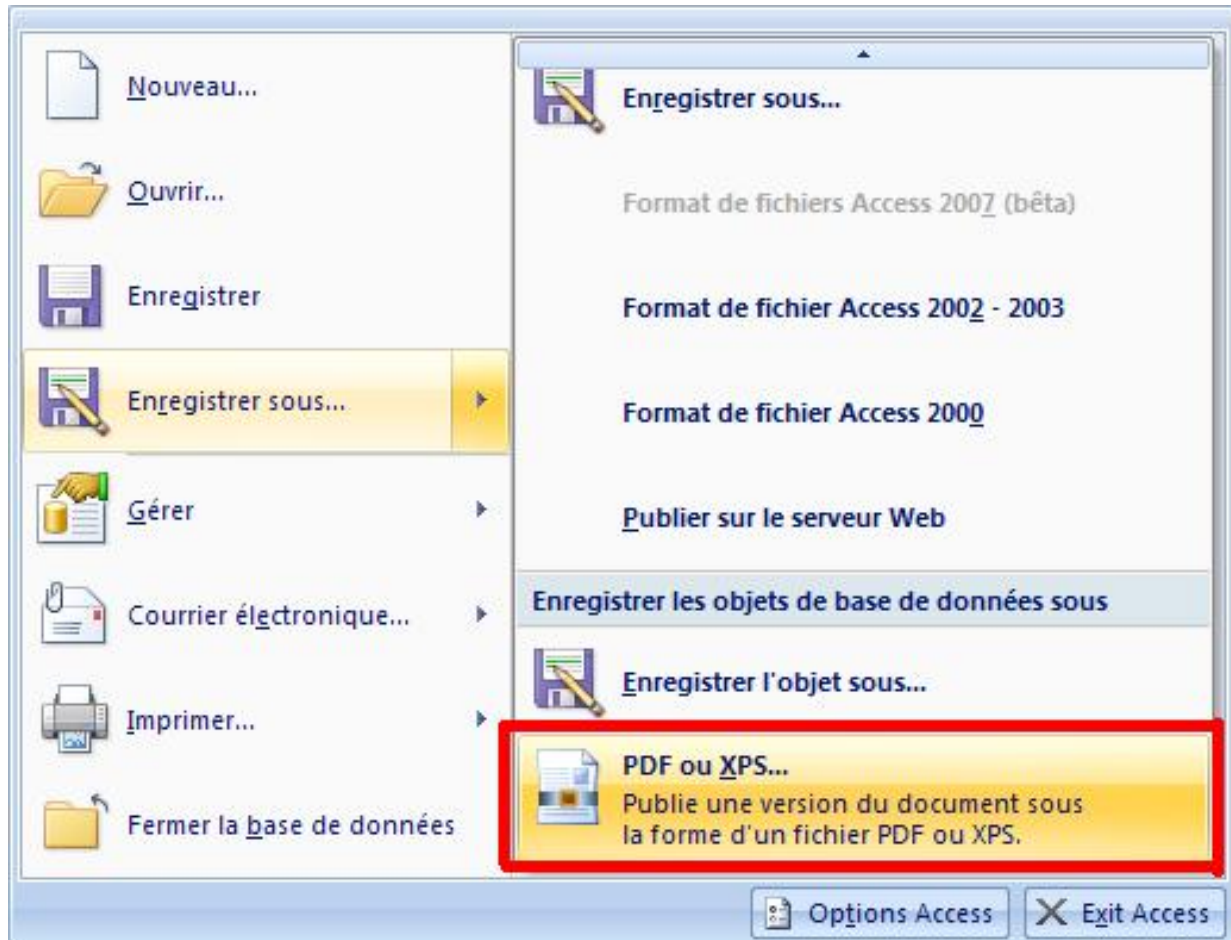
II-A - Vérification

Pour vérifier que l'extension PDF a été correctement installée, cliquez sur le menu **Fichier** (logo Office) puis sélectionnez la commande **Enregistrer Sous**.



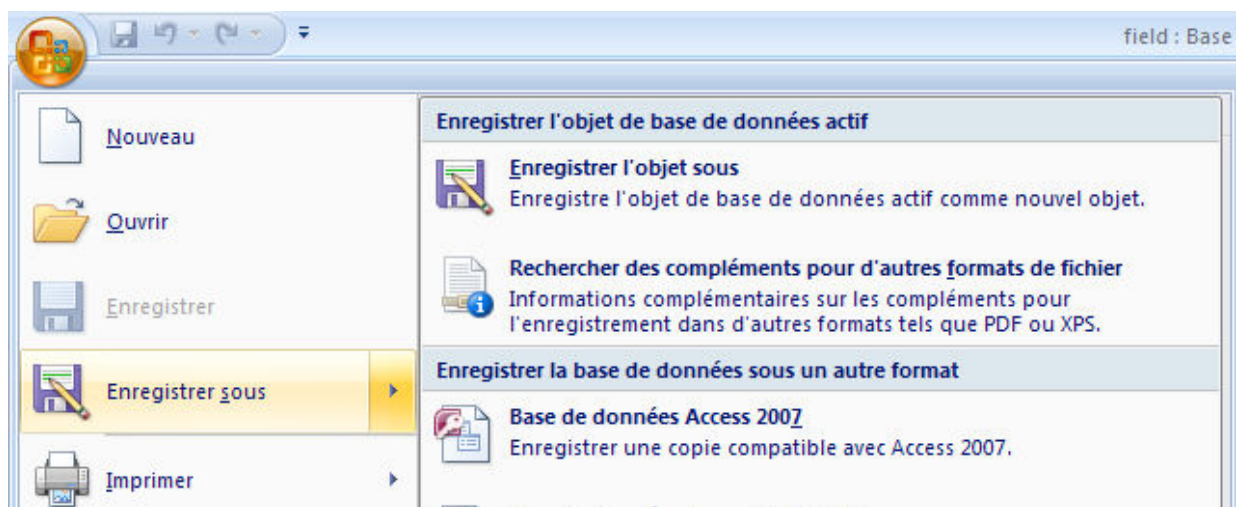
Cliquez ensuite sur la flèche de droite (elle permet d'afficher le sous-menu **Enregistrer sous**).

Vérifiez que la ligne **PDF ou XPS** est disponible. Dans le cas contraire, cela signifie que l'option PDF n'est pas installée.



II-B - Installation de l'extension

Il s'agit en fait d'un complément Office publié par Microsoft. Toujours dans le menu **Enregistrer Sous**, cliquez sur le bouton **Recherche des compléments pour d'autres formats de fichier**.



Une fenêtre de votre navigateur internet s'ouvre et vous amène sur le site de téléchargement de Microsoft.

Sélectionnez la langue désirée du produit puis cliquez sur **Continuer** et enfin **Télécharger**.

Continuer Validation requise

Ce téléchargement est disponible pour les utilisateurs de [Microsoft Office authentique](#). Cliquez sur [confidentialité](#), Microsoft n'utilise pas les informations collectées lors de la validation pour vos

Détails rapides

Nom du fichier :	SaveAsPDFandXPS.exe
Version :	1
Date de publication :	14/11/2006
Langue :	Français
Taille du téléchargement :	935 Ko
Durée de téléchargement estimée :	Accès distant (56 K) 3 min

Choisir une autre langue : Français

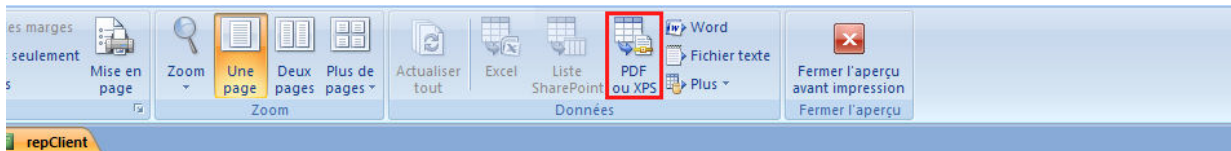
Enregistrez le fichier sur votre disque puis lancer l'installation en prenant garde de fermer vos applications Office auparavant.

Relancez Access. En théorie, les formats de fichiers XPS et PDF apparaissent dans le menu **Enregistrer Sous**.

III - Enregistrer un état au format PDF

III-A - Manuellement

Rien de plus simple, une fois l'état ouvert en mode **Aperçu avant impression**, un bouton **PDF ou XPS** apparaît. Cliquez dessus pour enregistrer l'état dans un fichier PDF.



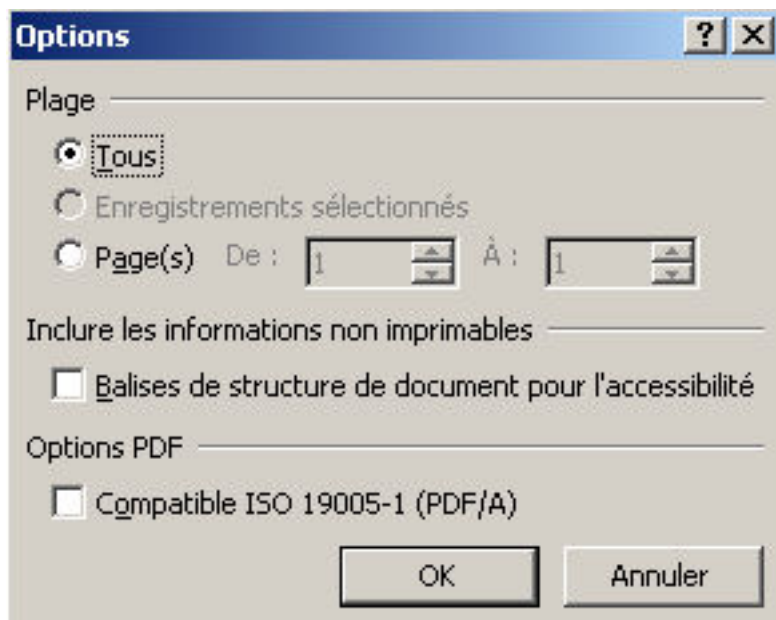
jeudi 12 octobre 2006



Liste des clients

Numéro	Nom	Prénom	Adresse	CP	Ville
1	MARTIN	Christophe	75 Rue du chaudron en bois	71000	MACON
2	MARTIN	Aurore	75 Rue du chaudron en bois	71000	MACON

La commande **Options** dans la boîte de dialogue **Enregistrer Sous** vous permet de régler quelques paramètres :



III-B - En VBA

Pour convertir un état en PDF avec VBA, deux méthodes peuvent être employées.

D'une part, la simple sauvegarde est réalisée en deux étapes :

- 1 Appel de **DoCmd.OutPutTo**
- 2 Ouverture de l'état en prévisualisation à l'aide de la méthode **DoCmd.OpenReport**

Dans l'exemple suivant, seule la page contenant le client numéro 2 est enregistré dans le fichier d:\test.pdf

```
DoCmd.OpenReport "repClient", acViewPreview, , "NumClient=2"  
DoCmd.OutputTo acOutputReport, , "PDF", "d:\test.pdf"
```

En fixant le paramètre **AutoStart** à **True**, le lecteur de PDF par défaut sera ouvert et affichera le document généré.

```
DoCmd.OutputTo acOutputReport, , "PDF", "d:\test.pdf", True
```



*Pour convertir au format XPS, il suffit de remplacer **PDF** par **XPS***

D'autre part, il est possible d'envoyer l'état en pièce-jointe d'un email. En une seule commande, VBA permet à la fois de créer l'email, de convertir l'état en PDF et de l'ajouter au document électronique.


Pour cela il faut utiliser la méthode **SendObject** de l'objet **DoCmd**. Attention, si l'utilisateur interrompt l'envoi, une erreur **2501** (la méthode **Docmd.SendObject** a échouée) est levée. Vous devez l'intercepter.

```
On Error GoTo err  
DoCmd.SendObject acSendReport, "repClient", _  
    "PDF", "destinataire@fai.com", , , "Liste des clients", _  
    "Veuillez trouver la liste des clients en pièce-jointe"  
  
fin:  
Exit Sub  
  
err:  
If err.Number <> 2501 Then err.Raise err.Number, , err.Description  
Resume fin
```

De la même manière que pour la sauvegarde, vous pouvez ouvrir l'état en mode prévisualisation pour filtrer sa source.

```
DoCmd.OpenReport "repClient", acViewPreview, , "NumClient=2"  
DoCmd.SendObject acSendReport, , _  
    "PDF", "destinataire@fai.com", , , "Liste des clients", _  
    "Veuillez trouver le détail du client n 2 en pièce-jointe."
```

IV - Créer un bouton dans le ruban

 Pour que cet exemple fonctionne vous devez ajouter les références :

- Microsoft Scripting Runtime
- Microsoft Office 12 Object Library

Prenons le cas d'un formulaire listant les clients d'une société.



Le but est de modifier le ruban de Microsoft Access de telle sorte qu'il affiche un onglet **Exporter** avec un bouton **PDF** chargé d'enregistrer l'état correspondant au client en cours.

Afin d'améliorer l'ergonomie, la méthode de sauvegarde doit permettre à l'utilisateur de choisir l'emplacement du fichier. Ceci peut être réalisé en utilisant les nouveaux objets [FileDialog](#).


```
Function PDF(ByVal control As IRibbonControl)
Dim oFD As Office.FileDialog
Dim strchemin As String
Set oFD = Application.FileDialog(msoFileDialogSaveAs)

With oFD
    'Paramétrage de la boîte de dialogue
    'Affichage en mode liste
    .InitialView = msoFileDialogViewList
    'Nom du fichier par défaut
    .InitialFileName = "Client" & Forms("frmClient").NumClient & ".pdf"
    'Titre de la fenêtre
    .Title = "Exporter en PDF"
    'Affiche la boîte et récupère le chemin du fichier
    If .Show Then
        If .SelectedItems.count > 0 Then
            strchemin = .SelectedItems(1)
            'Ouvre l'état
            DoCmd.OpenReport "repClient", acViewPreview, , "NumClient=" &
```

```

).NumClient
    'Enregistre l'état
    DoCmd.OutputTo acOutputReport, , "PDF", strchemin
    'Ferme l'état
    DoCmd.Close acReport, "repClient", acSaveNo
    End If
End If
End With
End Function

```

 Le paramètre **control As IRibbonControl** est nécessaire afin que la fonction soit valide avec le squelette du ruban. Vous trouverez de nombreuses explications dans le tutoriel :

[manipulation du ruban d'Access 2007 en VBA](#)

Etant donné que la fonction **PDF** sera exécutée par le ruban, elle sera placée dans un module spécifique : **mduRibbon**.

La définition de la structure du ruban est écrite dans un fichier XML : **ribbon.xml**. Pour simplifier le déploiement et la maintenance, veillez à toujours placer ce fichier dans le même répertoire que l'application. Son chargement sera assuré au démarrage de l'application (via une macro **AutoExec**) par la fonction suivante (module **mduRibbon**) :

```

Public Function initRibbon()
Dim strXML As String
Dim oFso As New FileSystemObject
Dim oFtxt As TextStream
'Charge le fichier XML en mémoire
Set oFtxt = oFso.OpenTextFile(CurrentProject.Path & _
    "\ribbon.xml", ForReading)
'Récupère le contenu
strXML = oFtxt.ReadAll
'Charge le ruban personnalisé correspondant
Application.LoadCustomUI "rubanFormulaireClient", strXML
End Function

```

N'oubliez pas que pour lier le formulaire à son ruban, il faut fixer sa propriété **Nom du ruban** à **rubanFormulaireClient**.

Enfin, il ne reste plus qu'à écrire la description du ruban dans le fichier **ribbon.xml**. Pour ne pas disperser l'utilisateur, il est préférable de réutiliser les icônes de Microsoft Office tant que possible. Ici, il s'agit de l'image PDF ou XPS nommée **PublishToPdfOrEdoc**.

```

<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="false">
    <tabs>
      <tab id="tabExporter" label="Exporter" visible="true">
        <group id="grpPDF" label="PDF">

              <button
                id="btnPDF"
                label="Enregistrer en PDF"
                size="large"
                onAction="PDF"
                imageMso="PublishToPdfOrEdoc"
              />
            </group>

        </tab>
      </tabs>
    </ribbon>
  </customUI>

```

Résultat :

Enregistrer en PDF
PDF

>> frmClient

N° :

Nom :

Prénom : Né le :

Adresse :

Sexe
 Homme
 Femme

V - Conclusion

Il était temps que Microsoft offre enfin la possibilité aux développeurs d'intégrer le PDF facilement dans leurs applications. L'automatisation de cette opération en VBA est très simple et son intégration dans le ruban peut répondre à 90 % des demandes. Les 10 % restant concernent la personnalisation du fichier généré (auteur, titre, etc.), ce qui n'est malheureusement pas pris en charge à l'heure actuelle. Espérons que Microsoft ira vers cet objectif dans les versions à venir (voire même avec un complément amélioré). Enfin, même s'il n'est pas possible de parler de réelle innovation, il n'en reste pas moins que cette nouveauté est un atout majeur d'Office 2007.

Je tiens à remercier l'équipe Access de www.developpez.com pour leur relecture.