

Tutoriel pour gérer une association avec Access

Comptabiliser les dons reçus en espèces ou par banque et produire les attestations fiscales

Par Claude Leloup 

Date de publication : 20 avril 2015

Cette application comportera :

- un formulaire pour enregistrer les coordonnées des différents intervenants ;
- un formulaire pour comptabiliser les opérations contre espèces et un autre pour les opérations via la banque ;
- un processus pour l'envoi des attestations fiscales par la poste ou par e-mail ;
- un processus pour l'envoi ciblé de courrier ;
- quelques états.

Vos commentaires sont les bienvenus dans cette discussion du forum **Commentez**

I - Prérequis.....	3
II - Quelles sont les données à gérer ?.....	3
II-A - Les acteurs intervenant dans l'association.....	3
II-B - Les flux financiers.....	4
III - Des formulaires pour peupler les tables.....	5
III-A - fContacts.....	5
III-B - fBanque et fCaisse.....	8
III-C - Quelques commentaires techniques.....	10
III-C-1 - Les images ne sont pas dans la base de données.....	10
III-C-2 - Se positionner sur un enregistrement précis.....	11
III-C-3 - La hauteur des sous-formulaires s'adapte suivant le nombre d'enregistrements.....	12
IV - Confection des attestations.....	13
IV-A - Le défi.....	13
IV-B - Le modèle de base.....	14
IV-C - Un petit clic, mais un paquet de code.....	16
IV-D - Construction d'une table tProvisoire pour le publipostage.....	16
IV-D-1 - La requête rCreatProvisoire.....	16
IV-D-2 - On ajoute une colonne NuméroAuto à la table tProvisoire.....	19
IV-D-3 - À ce stade, voici à quoi ressemble tProvisoire.....	19
IV-E - Une procédure pour publiposter les lettres qui seront mises sous pli.....	19
IV-E-1 - Les étapes.....	20
IV-E-2 - Le code.....	20
IV-F - Une procédure pour publiposter des lettres qui seront envoyées par e-mail.....	23
IV-F-1 - Les étapes.....	24
IV-F-2 - Le code.....	26
V - Archivage des attestations fiscales.....	31
VI - Envoi de courrier ciblé.....	32
VI-A - Le formulaire fEnvoiCourrier.....	32
VI-B - Le code.....	34
VII - Quelques états.....	38
VII-A - Un premier choix.....	38
VII-B - Le code associé aux listes présentées ici.....	40
VIII - Deux bases : une dorsale et une frontale.....	42
IX - Une arborescence particulière.....	42
IX-A - Adaptation automatique de la liaison dorsale/frontale, si changement de localisation.....	43
X - Téléchargement.....	45
XI - Remerciements.....	45

I - Prérequis

Ce tutoriel s'adresse à des utilisateurs qui maîtrisent déjà les bases du logiciel Access.

Pour vérifier votre niveau, parcourez ces tutoriels : si vous les comprenez facilement, vous êtes OK et si ce n'est pas le cas, insistez :

- pour commencer : **Maxence Hubiche Access - Les Bases** ;



- pour construire des requêtes : **Jean Ballat Crée des requêtes simples** ;

- pour construire un formulaire : **Jean-Philippe Ambrosino** le chapitre 2-1-2 de **Mise en surbrillance d'un enregistrement dans un formulaire** ;

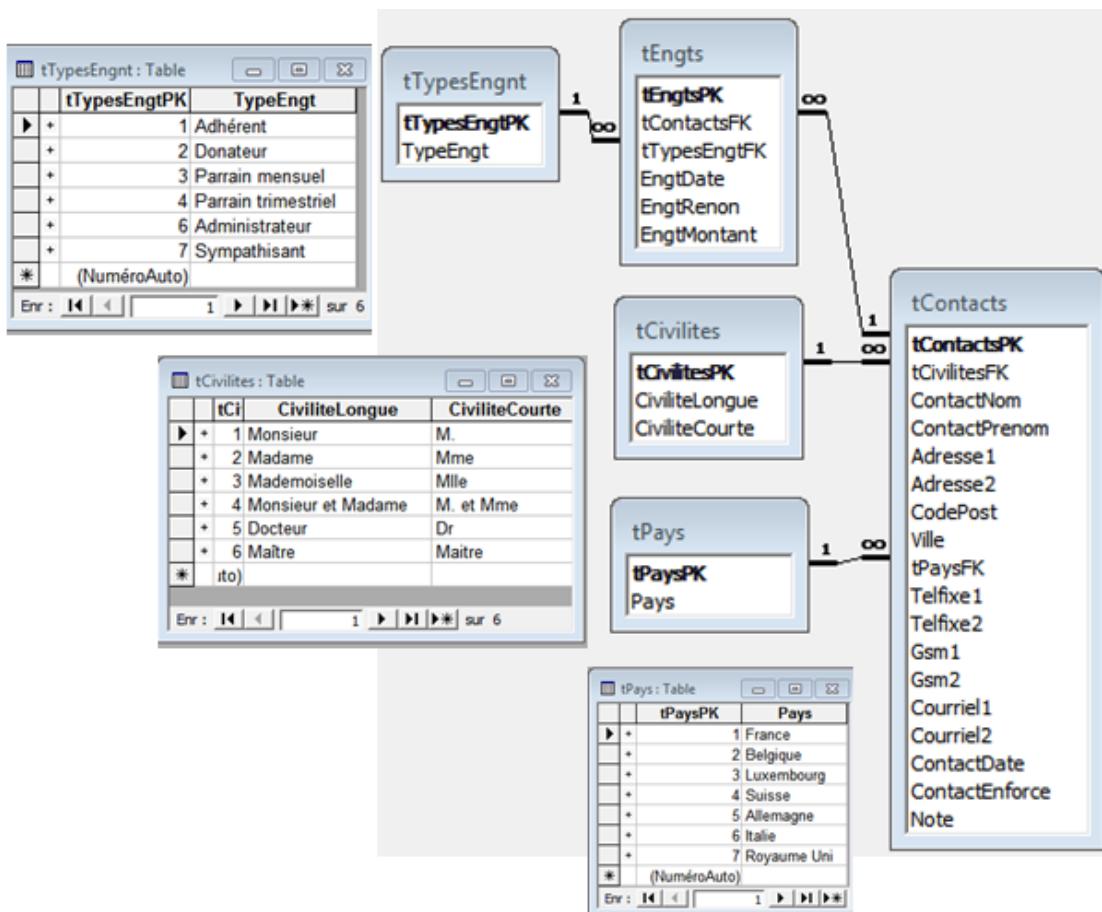
- pour le VBA : **Olivier Lebeau Initiation au VBA Office**.

II - Quelles sont les données à gérer ?

II-A - Les acteurs intervenant dans l'association

- Leurs coordonnées classiques : identité, adresse (pour leur faire suivre l'attestation fiscale, envoyer de l'info...).
- Leur rôle : donateur, parrain, administrateur... (pour leur envoyer des infos ciblées).
- Leurs engagements (quoi, depuis quand, jusqu'à quand).

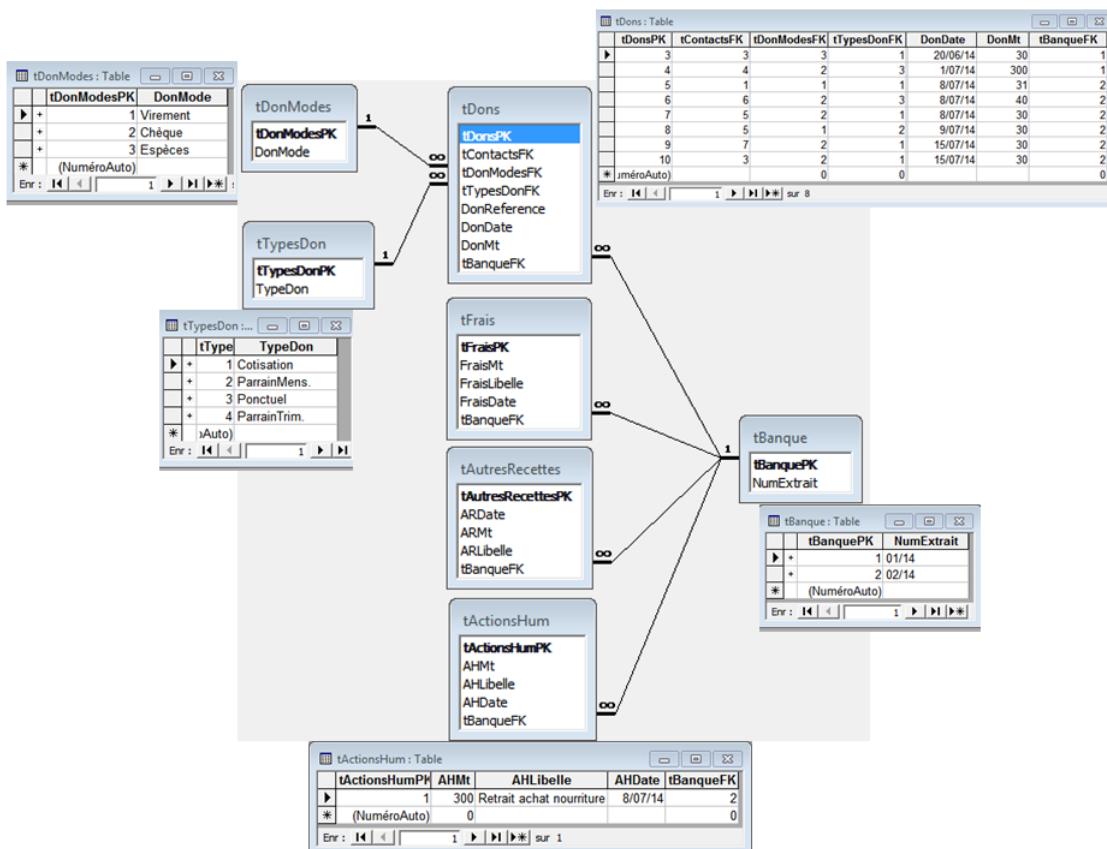
Pour donner une idée :



II-B - Les flux financiers

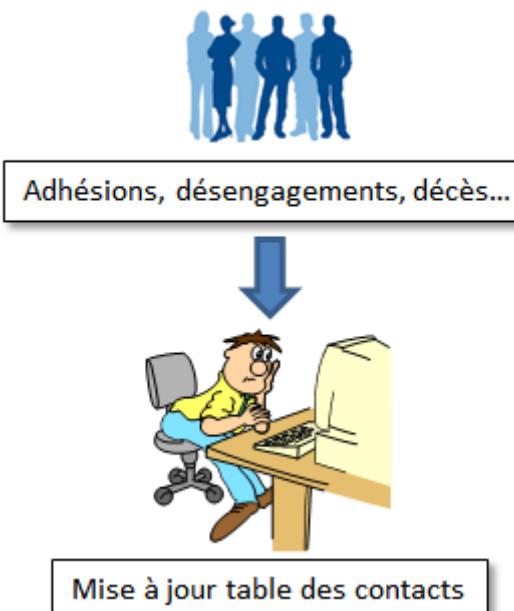
- D'où vient l'argent : qui a donné ? Quand ? Et pour quoi faire ? (Cotisations, parrainage, marchés de Noël, soirées spaghetti...)
- Ce qu'on a fait de l'argent collecté : quand ? Pour quel objet ? (Aide aux démunis, frais de fonctionnement divers...)

Pour donner une idée :



III - Des formulaires pour peupler les tables

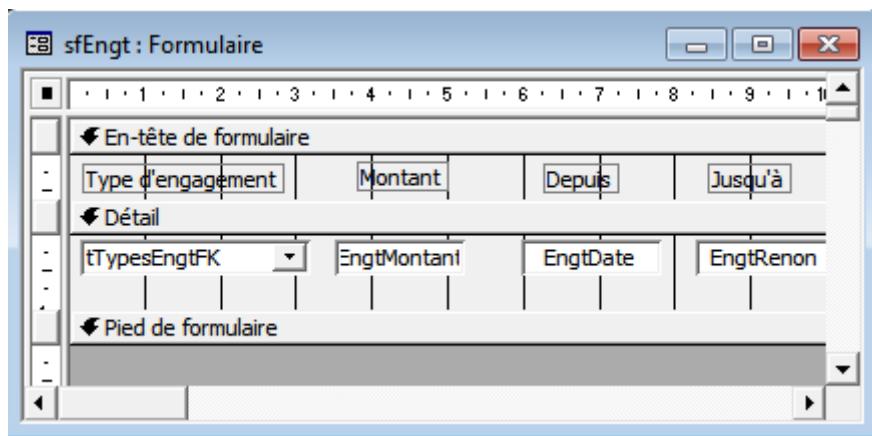
III-A - fContacts



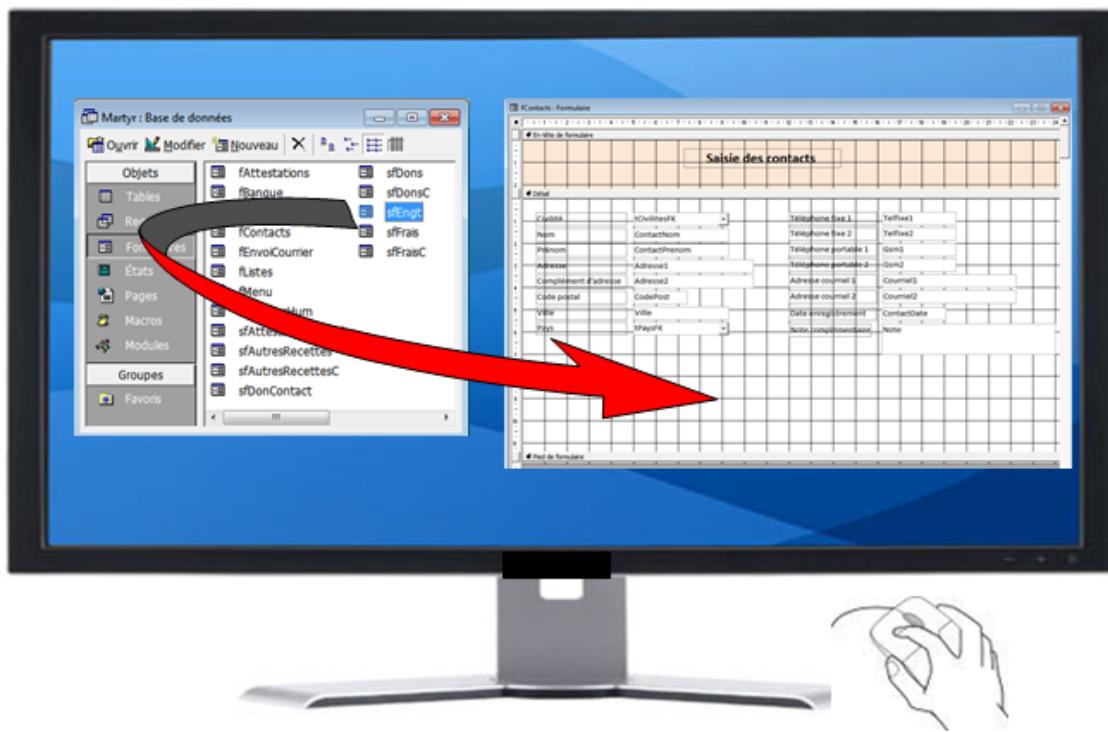
- Dans un premier temps avec comme source *tContacts*



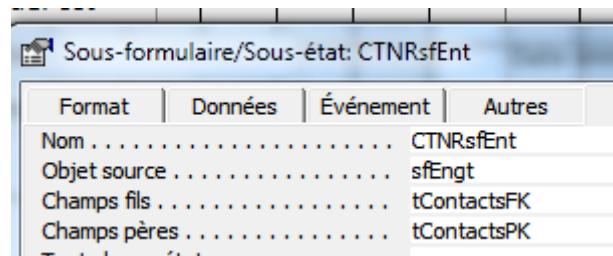
- Parallèlement, un formulaire *sfEngt* avec comme source *tEngts*



- On glisse et dépose le second dans le premier



et comme les tables sont liées entre elles, on obtient un formulaire père/fils



 Si cette technique ne vous est pas familière, voyez ce tutoriel [Comment classer les données dans des tables liées et construire un formulaire père/fils](#)

Et nous obtenons ceci :

F_Contacts

Saisie des contacts

Afficher :

Civilité	Monsieur	Téléphone fixe 1	03 89 90 91 92
Nom	Tell	Téléphone fixe 2	
Prénom	Guillaume	Téléphone portable 1	06 78 90 11 12
Adresse	50 rue du Ri	Téléphone portable 2	
Complément d'adresse	Résidence Rossini	Adresse courriel 1	guillaume.tell@pomme.ch
Code postal	1014	Adresse courriel 2	
Ville	Lausanne	Date enregistrement	1/10/12
Pays	Suisse	Note complémentaire	

Ses engagements			
Type d'engagement	Montant	Depuis	Jusqu'à
Administrateur	€ 0,00	15/08/13	
Adhérent	€ 30,00	20/06/14	31/12/14
*	€ 0,00		

Date	Type don	Montant	Compta
20/06/14	Cotisation	30,00	fCaisse

Enr : 2 sur 8

N.B. L'image anticipe un peu : le sous-formulaire fils *sfDonContact* sera ajouté plus tard en usant de la même technique.

III-B - fBanque et fCaisse



L'idée est de concevoir des formulaires :

- qui permettent de saisir les entrées et sorties de fonds en les classant par nature (cotisations reçues, octrois d'aide financière, frais de fonctionnement...) comme dans une comptabilité en partie double ;

- qui incluent un système d'autocontrôle. Par exemple, si le formulaire *fBanque* restitue le cumul des entrées et sorties qui y sont enregistrées et que cette somme correspond au solde renseigné à l'extrait délivré par la banque, il y a de fortes chances que l'encodage soit complet.

N° que porte l'extrait bancaire
 (N°/Année)

N° Extrait **02/14**

Solde précédent : **390,00**
 + dons reçus : **191,00**
 + autres recettes : **320,00**
 - frais divers : **-10,00**
 - actions humanitaires : **-300,00**
 = solde à l'extrait : **591,00**

Dons et cotisations

Date	Donateur	Type don	Mode	Références	Montant
8/07/14	Desautres Jean et Monique	Cotisation	Chèque		30,00
8/07/14	Lerouge Daniel	Ponctuel	Chèque		40,00
8/07/14	UneTelle Emillenne	Cotisation	Virement		31,00
9/07/14	Desautres Jean et Monique	ParrainMens.	Virement		30,00
15/07/14	Lesbains François	Cotisation	Chèque		30,00
15/07/14	Debruges Jef	Cotisation	Chèque		30,00
*					
Nbre postes : 6					Total : 191,00

Frais divers / retraits

Date	Libellé	Montant
1/07/15	frais tenue de compte	10,00
*		0,00
Nbre postes : 1		Total : 10,00

Actions humanitaires

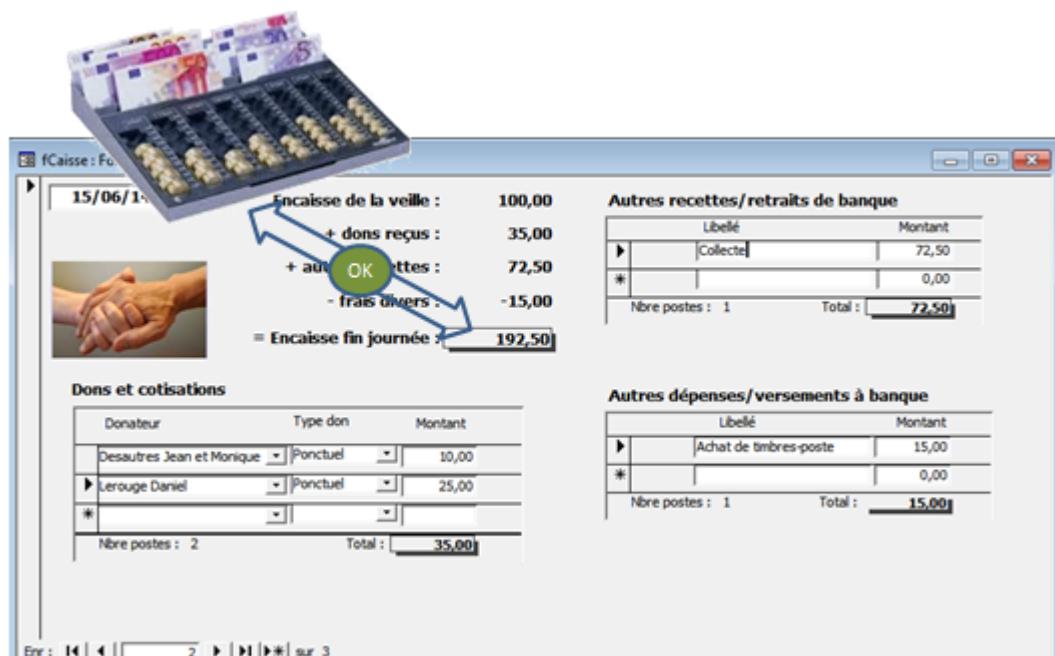
Date	Libellé	Montant
8/07/14	Achat nourriture Cambodge	300,00
*		0,00
Nbre postes : 1		Total : 300,00

Des sous-formulaires « fils »
 pour encoder les différentes catégories de mouvements.
 Leur taille s'adapte au nbre de postes (ascenseur si besoin).

Enr : **2** sur 2

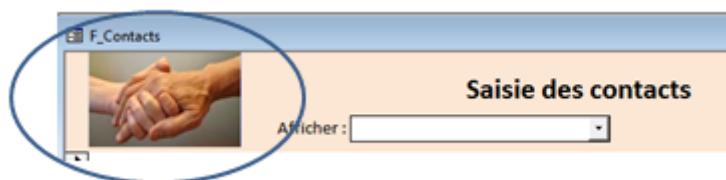
Données calculées par le système.
 Le fait que le solde calculé corresponde avec le solde de la banque augure que l'on a bien tout reporté.

De même pour les espèces en caisse



III-C - Quelques commentaires techniques

III-C-1 - Les images ne sont pas dans la base de données



Elles sont logées dans un sous-répertoire IMAGES.

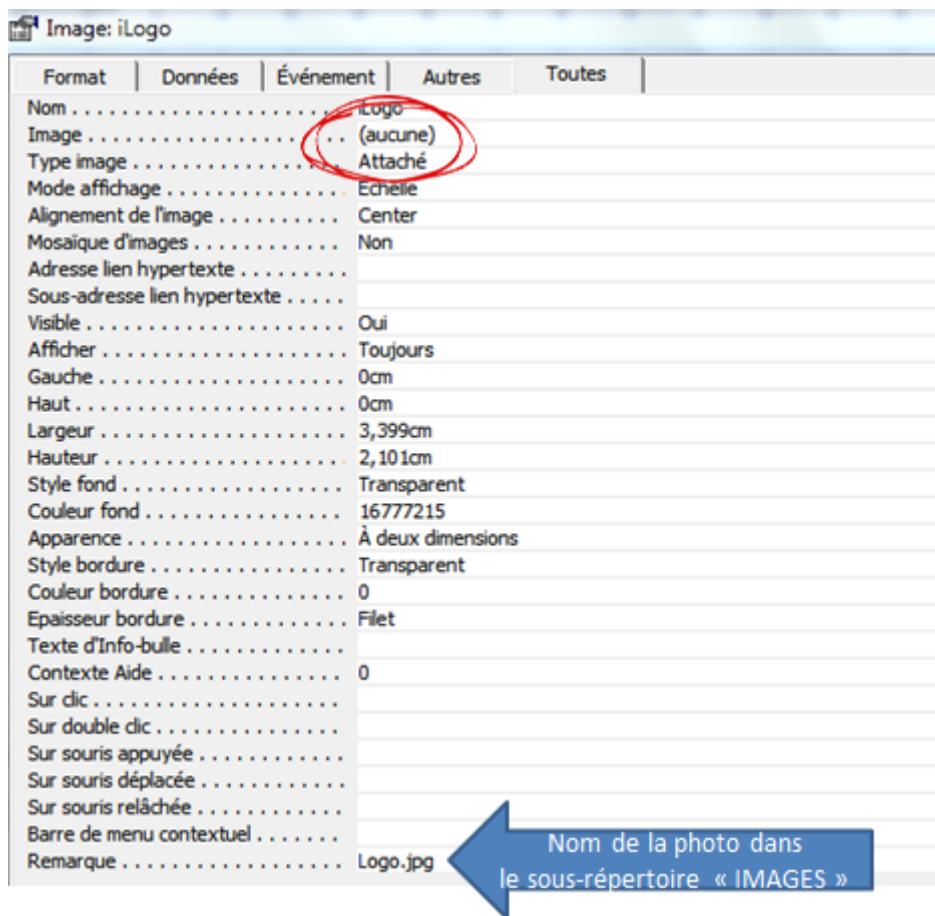
À l'ouverture du formulaire, ce code

```
Private Sub Form_Open(Cancel As Integer)
    Call AmnImages
End Sub
```

La routine appelée est logée dans le module *m/Images*.

Le principe est le suivant :

- le contrôle image a cette structure



- à l'ouverture du formulaire (ou de l'état), la routine *AmnImages* repère tous les contrôles de type image dans le formulaire et adapte leur propriété *Image* (Picture) en modifiant le chemin de l'image à afficher

Extrait de la routine AmnImages

```
For Each Ctrl In CodeContextObject.Controls
    If Ctrl.Picture = "(aucune)" And Ctrl.PictureType = 1 And Ctrl.Tag Like "*.*" Then
        Ctrl.Picture = CurrentProject.Path & "\Images\" & Ctrl.Tag
    End If
Next Ctrl
```

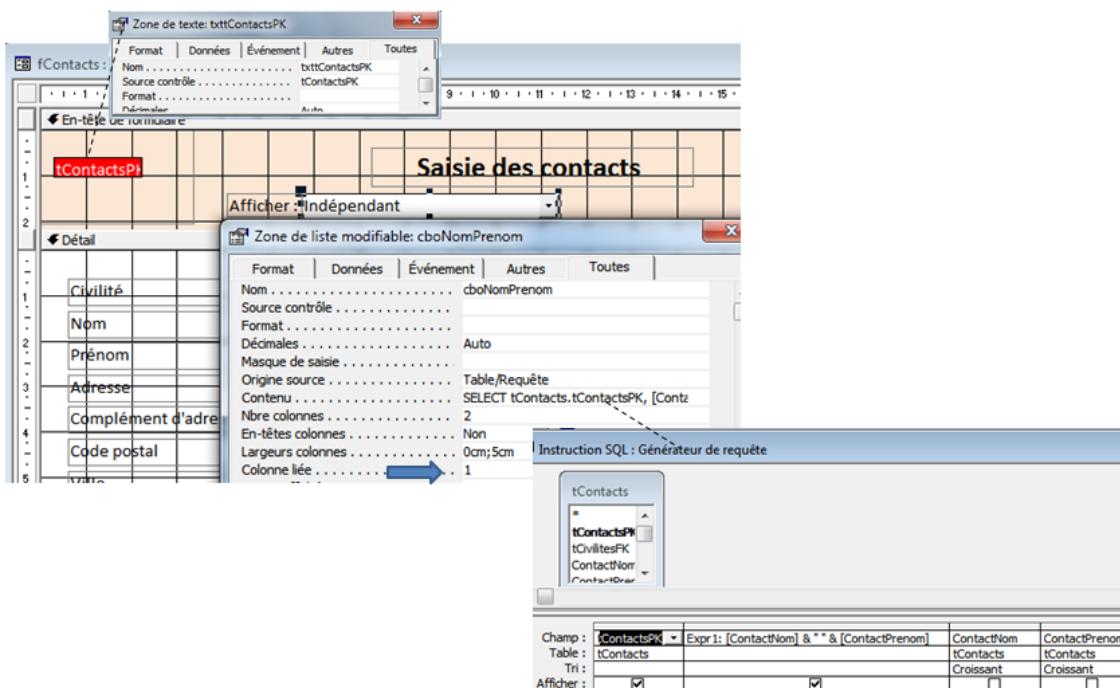
Vous trouverez plus de détails sur cette routine dans cet autre tutoriel : **Stockez les images statiques de vos formulaires et états Access hors de la base de données.**

III-C-2 - Se positionner sur un enregistrement précis



Civilité	Afficher :	Télépho
Nom	Debruges Jef	Télépho
Prénom	Desautres Jean et Monique	Télépho
Adresse	Encorun Bienvenu	Télépho
Complément d'adresse	Lerouge Daniel	Télépho
	Lesbains François	Télépho
	Mouton Bernard et Françoise	Télépho
	Tell Guillaume	Télépho
	UneTelle Emilienne	Adresse

Quand l'utilisateur a choisi un contact dans la liste, le contrôle *cboNomPrenom* contient la clé de ce contact *tContactsPK*



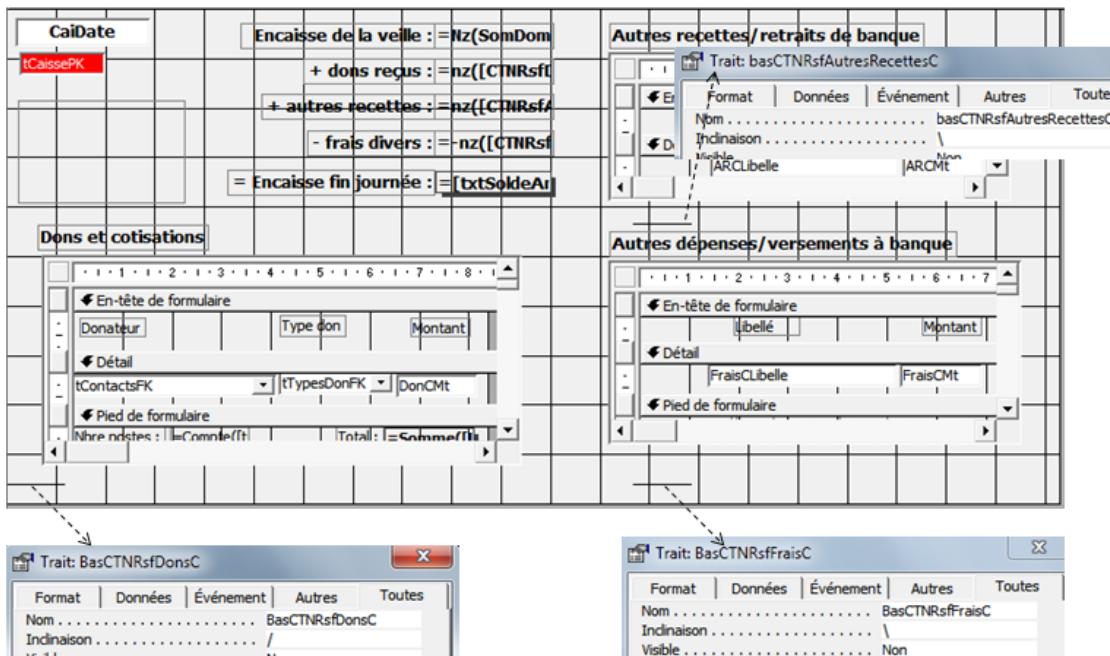
Pour se positionner sur l'enregistrement choisi, remettre la zone de liste modifiable à Null et donner le focus sur le nom et prénom :

```
Private Sub cboNomPrenom_AfterUpdate()
    DoCmd.GoToControl "txtContactsPK"
    DoCmd.FindRecord Me.cboNomPrenom
    Me.cboNomPrenom = Null
    Me.txtContactNom.SetFocus
End Sub
```

III-C-3 - La hauteur des sous-formulaires s'adapte suivant le nombre d'enregistrements

Ceci pour que la ligne « Total » colle à la dernière ligne du détail.

En bref, la hauteur du conteneur dépend du nombre enregistrements du formulaire fils. On place dans le formulaire « père » une limite au-delà de laquelle on veut stopper la progression. Une barre de défilement verticale se met alors en place.



Le processus d'adaptation est déclenché à chaque affichage d'un enregistrement du père

```
Private Sub Form_Current()
    'Moduler la taille des sous-formulaires
    Call AmenagerTailleSF(Me.Name, "CTNRSfDonsC")
    Call AmenagerTailleSF(Me.Name, "CTNRSfAutresRecettesC")
    Call AmenagerTailleSF(Me.Name, "CTNRSfFraisC")
    Me.txtCaiDate.SetFocus
End Sub
```

et à chaque ajout ou suppression d'un enregistrement dans le fils

```
Private Sub Form_AfterDelConfirm(Status As Integer)
    Call AmenagerTailleSF(Me.Parent.Name, Me.Parent.ActiveControl.Name)
    Me.Requery
End Sub

Private Sub Form_AfterInsert()
    Call AmenagerTailleSF(Me.Parent.Name, Me.Parent.ActiveControl.Name)
    Me.Requery
End Sub
```

La procédure *AmenagerTailleSF()* est logée dans le module *mTailleSF*.

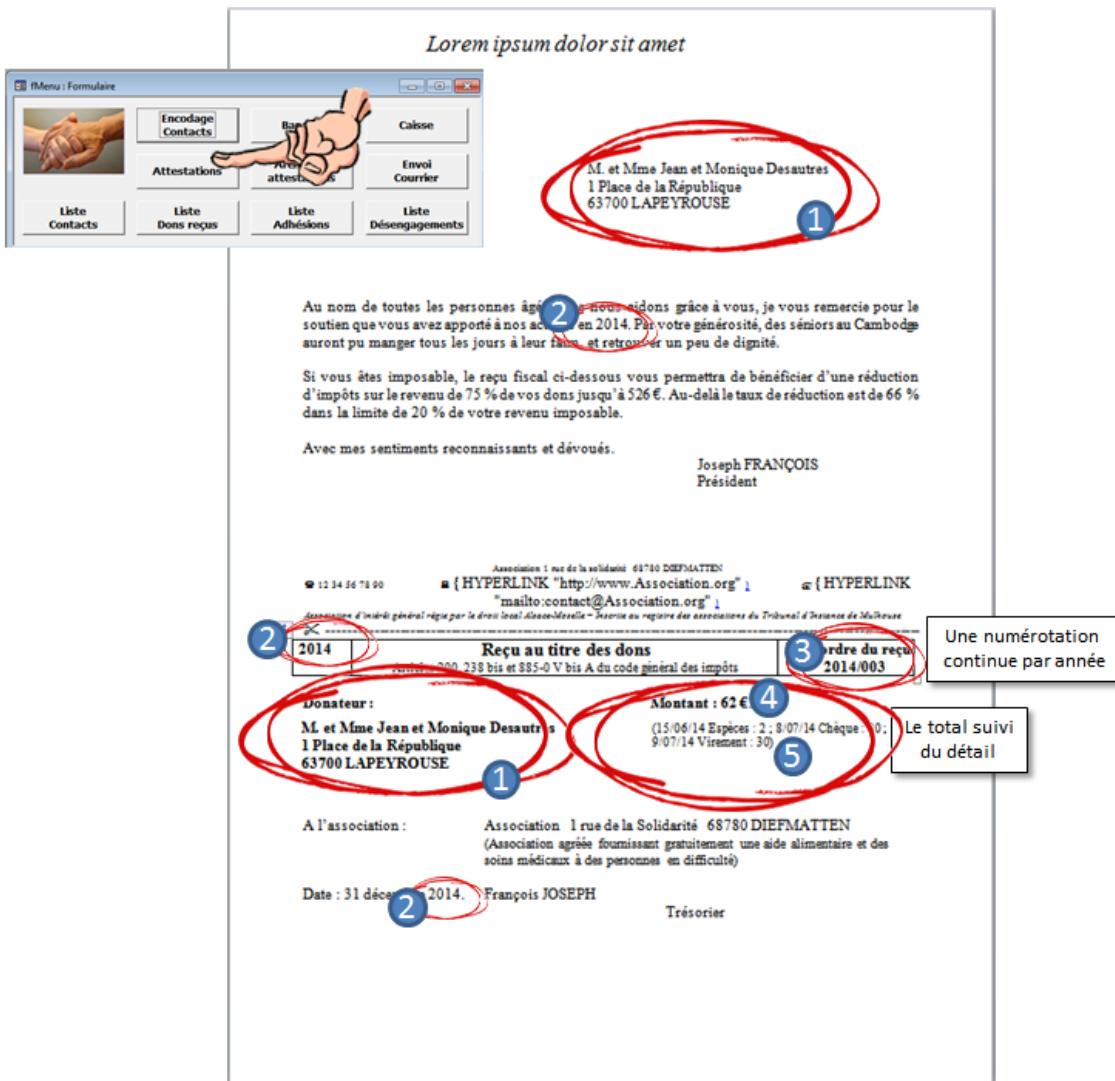
Un commentaire détaillé de ce code peut être consulté dans [ce tutoriel](#).

IV - Confection des attestations

IV-A - Le défi

En un clic sur un bouton, les attestations fiscales de l'année sont confectionnées,
- sur papier pour chaque donateur sans adresse e-mail ;

- sous forme d'un PDF, joint à un mail pour les autres.



Nous allons piloter un publipostage Word depuis Access.

IV-B - Le modèle de base

Voici comment il se présentera :

Lorem ipsum dolor sit amet

1 {MERGEFIELD "Donateur"}

Au nom de toutes les personnes âgées que nous aidons grâce à vous, je vous remercie pour le soutien que vous avez apporté à nos actions. 2 {MERGEFIELD "Annee"}. Par votre générosité, des séniors au Cambodge auront pu manger tous les jours à leur faim, et retrouver un peu de dignité.

Si vous êtes imposable, le reçu fiscal ci-dessous vous permettra de bénéficier d'une réduction d'impôts sur le revenu de 75 % de vos dons jusqu'à 526 €. Au-delà le taux de réduction est de 66 % dans la limite de 20 % de votre revenu imposable.

Avec mes sentiments reconnaissants et dévoués.

Joseph FRANÇOIS
Président

Association 1 rue de la solidarité 68780 DIEFMATTEN
 ☎ 12 34 56 78 90 ■ {HYPERLINK "http://www.Association.org"}
 "mailto:contact@Association.org" ↗ {HYPERLINK
Association d'intérêt général régie par le droit local Alsace-Moselle - Inscriue au registre des associations du Tribunal d'Instance de Mulhouse

2 {MERGEFIELD "Annee" }	Reçu au titre des dons Articles 200, 238 bis et 885-0 V bis A du code général des impôts	3 {MERGEFIELD "NumOrdre" }
1 {MERGEFIELD "Donateur" }	4 Montant : {MERGEFIELD "Total" } €. 5 {MERGEFIELD "Detail" }	

À l'association : Association 1 rue de la Solidarité 68780 DIEFMATTEN
(Association agréée fournit gratuitement une aide alimentaire et des soins médicaux à des personnes en difficulté)

Date : 31 décembre 2 {MERGEFIELD "Annee" }. François JOSEPH
Trésorier



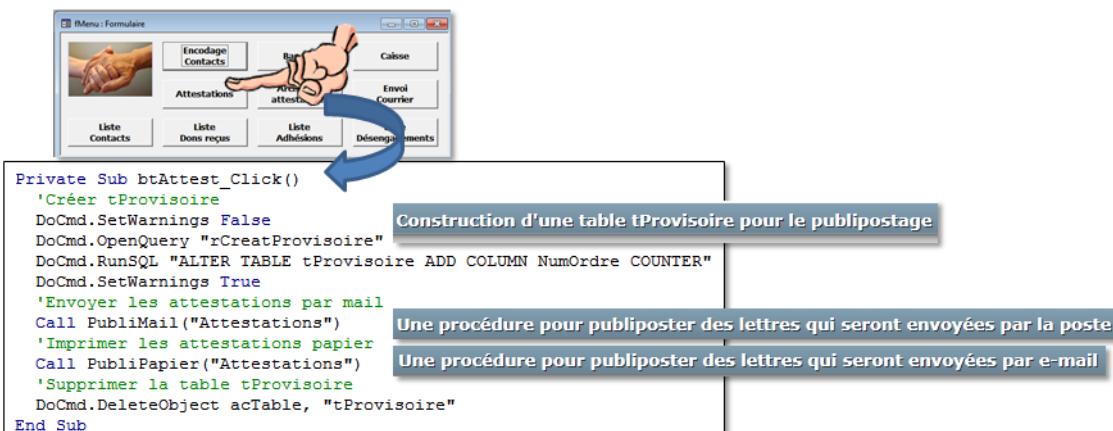
Si la technique ne vous est pas familière, voyez une description dans ce tutoriel.

Nous devons donc construire une table qui contiendra pour chaque donateur :

Nom de la colonne	Contenu
1 Donateur	4 lignes Civilité, prénom, nom Adresse Complément d'adresse éventuel

	Code postal, localité
Annee	L'année des dons 2
NumOrdre	Une numérotation continue 3
Total	Le total des dons 4
Detail	Pour chaque don La date, le mode, le montant 5

IV-C - Un petit clic, mais un paquet de code



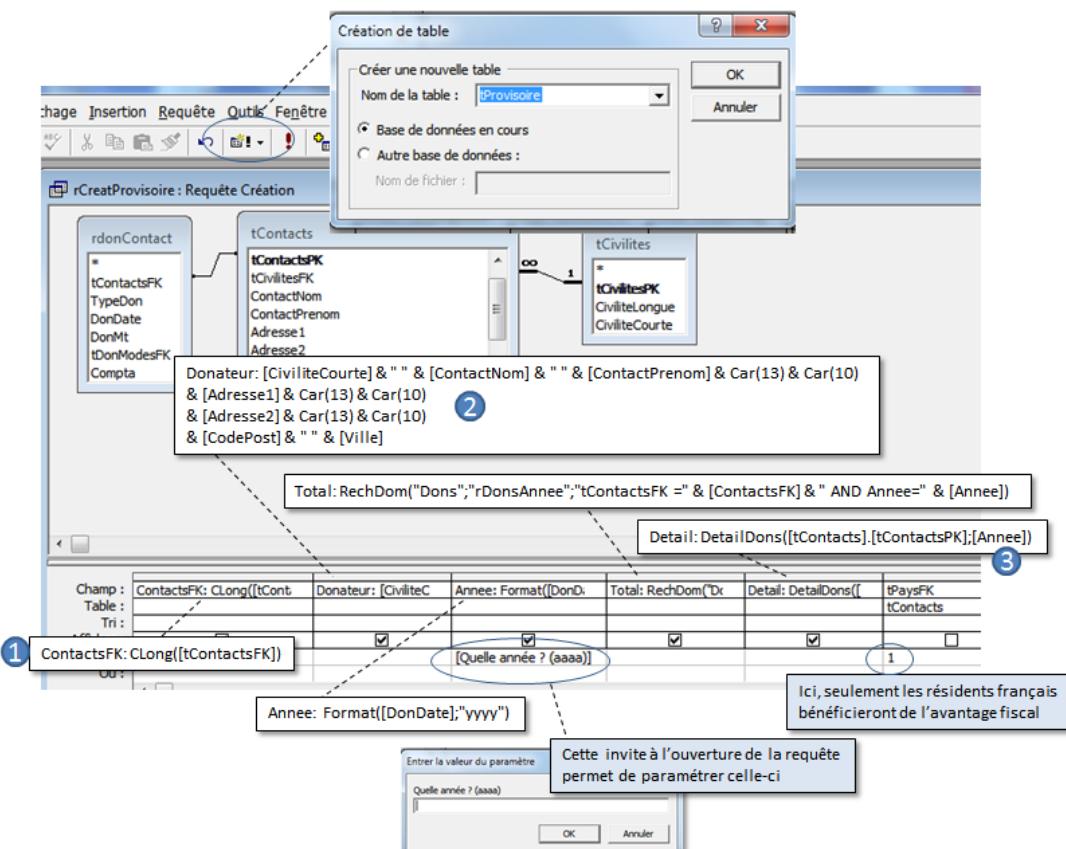
IV-D - Construction d'une table tProvisoire pour le publipostage

IV-D-1 - La requête rCreatProvisoire

```

SELECT DISTINCT CLng([tContactsFK]) AS ContactsFK, [CiviliteCourte] & " " & [ContactNom] &
" & [ContactPrenom] & Chr(13) & Chr(10) & [Adresse1] & Chr(13) & Chr(10) & [Adresse2] & Chr(13) & Chr(10) & [CodePostal] & " " & [Ville] AS Donateur, Format([DonDate],"yyyy") AS Annee,
DLookUp("Dons","rDonsAnnee","tContactsFK =" & [ContactsFK] & " AND Annee=" & [Annee]) AS Total,
DetailDons([tContacts].[tContactsPK],[Annee]) AS Detail INTO tProvisoire
FROM tCivilites INNER JOIN (rdonContact INNER JOIN tContacts ON rdonContact.tContactsFK =
tContacts.tContactsPK) ON tCivilites.tCivilitesPK = tContacts.tCivilitesFK
WHERE (((Format([DonDate],"yyyy"))=[Quelle année ? (aaaa)]) AND ((tContacts.tPaysFK)=1));

```



- 1 Cette transformation en **Long** pour éviter que la colonne créée soit NuméroAuto.
- 2 & Car(13) & Car(10) provoque un saut à la ligne lors de l'impression.
- 3 La fonction **Detail()**

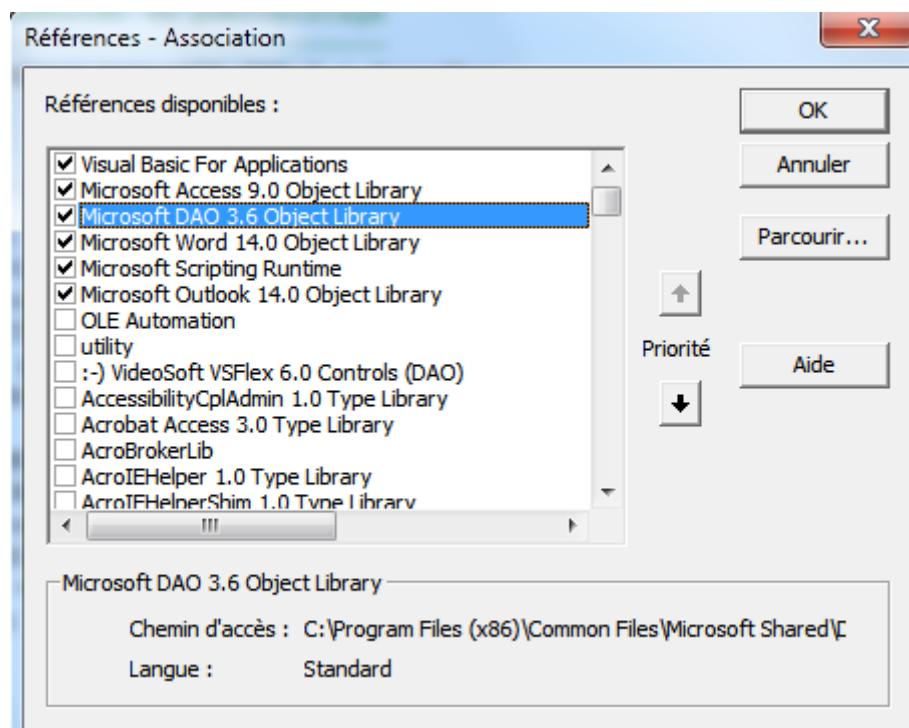
```

1. Public Function DetailDons(tContactsPK As Long, Annee As Integer) As String
2. Dim sSql As String
3. Dim rs As Recordset
4. sSql = "SELECT rdonContact.tContactsFK, Format([DonDate], ""yyyy"") AS Annee, "
5.     & "rdonContact.DonDate, tDonModes.DonMode, rdonContact.DonMt "
6.     & "FROM rdonContact INNER JOIN tDonModes "
7.     & "ON rdonContact.tDonModesFK = tDonModes.tDonModesPK "
8.     & "WHERE tContactsFK =" & tContactsPK & " And Format([DonDate], ""yyyy"") =" & Annee
9.     & " ORDER BY rdonContact.DonDate;"
10. Set rs = CurrentDb.OpenRecordset(sSql)
11. Do Until rs.EOF
12.     DetailDons = DetailDons & rs("DonDate") & " " & rs("DonMode") & " :
" & Format(rs("DonMt"), "#.00") & " "
13.     rs.MoveNext
14. Loop
15. End Function

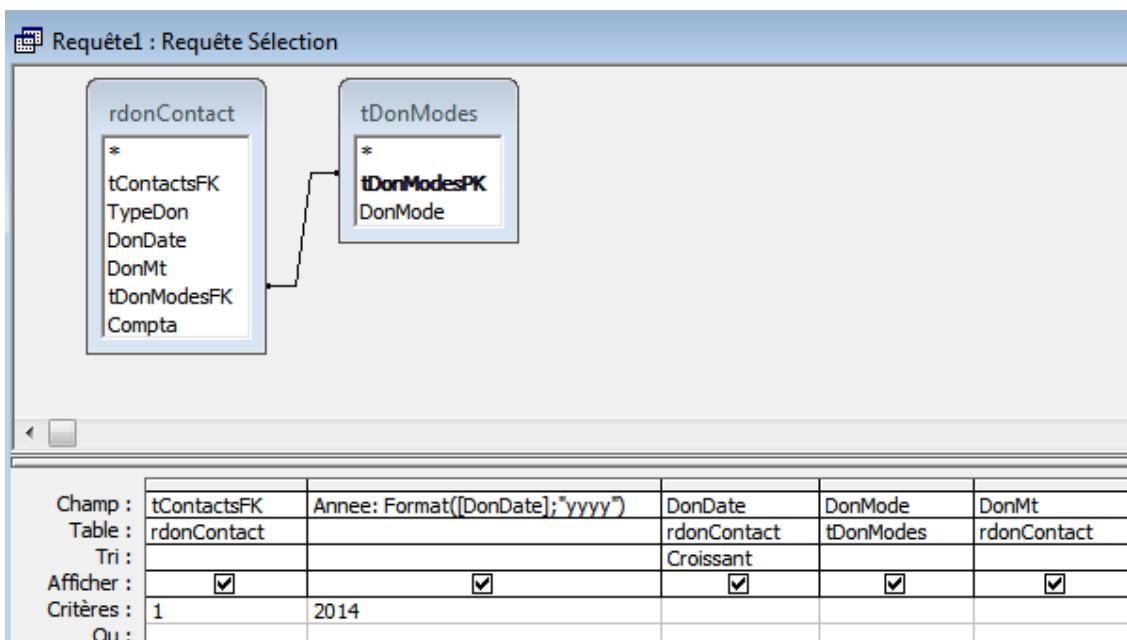
```

Explication du code

1-2 : la définition d'une variable de type Recordset implique d'ajouter la bibliothèque Microsoft DAO Object Library au projet



4-9 : on construit le SQL d'une requête qui aurait cet aspect si on appelait la fonction comme ceci **DetailDons(1,2014)**



donc une requête qui ramène tous les dons reçus de la part du donneur 1 pour l'année 2014.

10 : on crée un jeu d'enregistrements (*recordset*) avec ce SQL.

11-14 : on lit un à un les enregistrements et pour chacun (12), on concatène la date, le mode et le montant.

En l'occurrence, la fonction donnerait ceci :

Exécution

```
? DetailDons (1, 2014)
12/06/14 Espèces : 100,00 8/07/14 Virement : 31,00
```

IV-D-2 - On ajoute une colonne NuméroAuto à la table tProvisoire

Avec ce code

```
DoCmd.RunSQL "ALTER TABLE tProvisoire ADD COLUMN NumOrdre COUNTER"
```

IV-D-3 - À ce stade, voici à quoi ressemble tProvisoire

	ContactsFK	Donateur	Année	Total	Detail	NumOrdre
▶	3	M. et Mme Lesbains Fr	2014	60,55	20/06/14 Espèc	1
	4	M. et Mme Mouton Ben	2014	300	1/07/14 Chèque	2
	5	M. et Mme Desautres J	2014	70	15/06/14 Espèc	3
*	6	M. et Mme Lerouge Dar	2014	65	15/06/14 Espèc	4
					(NuméroAuto)	

IV-E - Une procédure pour publiposter les lettres qui seront mises sous pli

IV-E-1 - Les étapes



IV-E-2 - Le code

```

1. Public Sub Publipapier(Genre As String)
2. Dim objWord As Word.Application
3. Dim NomDoc As String
4. Dim ModeleDoc As String
5. Dim SavedDoc As String
6.
7. 'Ajuster les paramètres en fonction du Genre
8. -----
9.
10. Select Case Genre
11.
12. Case "Attestations"
13.     ModeleDoc = CurrentProject.Path & "\ATTESTATIONS\AttestationsModele.doc"
14.     SavedDoc = CurrentProject.Path & "\AEnvoyer\AttestationsAEnvoyer.doc"
15.     'Créer la table tPublipapier
16.     DoCmd.SetWarnings False
17.     DoCmd.RunSQL "SELECT tProvisoire.*, tContacts.Courriell INTO tPublipapier " _
        & "FROM tContacts INNER JOIN tProvisoire " _
18.             & "ON tContacts.tContactsPK = tProvisoire.ContactsFK " _
19.                 & "WHERE tContacts.Courriell Is Null;" _
20.
21.     DoCmd.SetWarnings True
22.
23. Case "ArchivAttest"
24.     DoCmd.SetWarnings False
25.     DoCmd.RunSQL "SELECT tProvisoire.*, tContacts.Courriell INTO tPublipapier " _
        & "FROM tContacts INNER JOIN tProvisoire " _
26.             & "ON tContacts.tContactsPK = tProvisoire.ContactsFK;" _
27.
28.     DoCmd.SetWarnings True
29.     ModeleDoc = CurrentProject.Path & "\ATTESTATIONS\AttestationsModele.doc"
30.     SavedDoc = CurrentProject.Path & "\ATTESTATIONS\Archives\" _
            & DLookup("Annee", "tPublipapier") & "ArchivageAttestations.doc"
31.
32.

```

```

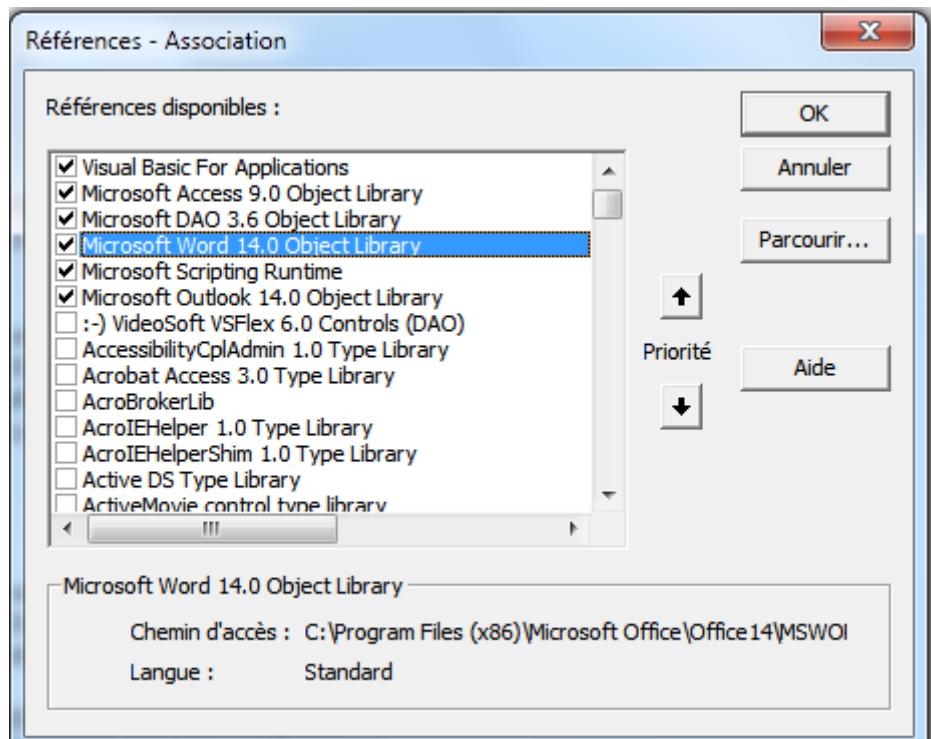
33. Case "Courrier"
34.     ModeleDoc = Forms!fEnvoiCourrier!txtCheminDoc
35.     SavedDoc = CurrentProject.Path & "\AEnvoyer\CourrierAEvoyeur.doc"
36.     DoCmd.SetWarnings False
37.     DoCmd.RunSQL "SELECT * INTO tPubliPapier " _
38.                 & "FROM tCourrier WHERE tCourrier.Poste=True;" 
39.     DoCmd.SetWarnings True
40.
41. End Select
42.
43. 'Publiposter
44. -----
45. Set objWord = New Word.Application
46. With objWord
47.     ' Ouvrir le document modèle
48.     .Documents.Open ModeleDoc
49.     .ActiveDocument.MailMerge.OpenDataSource _
50.         Name:=CurrentDb.Name, SQLStatement:="SELECT * FROM [tPubliPapier]"
51.     .ActiveDocument.MailMerge.Execute
52.     .ActiveDocument.SaveAs2 SavedDoc
53.     .Documents.Close
54. End With
55.
56. ' Fermer et libérer les objets
57. -----
58. objWord.Quit
59. Set objWord = Nothing
60.
61. 'Supprimer la table tPubliPapier
62. -----
63. DoCmd.DeleteObject acTable, "tPubliPapier"
64.
65. 'Montrer le résultat du publipostage
66. -----
67. Shell "C:\WINDOWS\EXPLORER.EXE " & SavedDoc
68.
69. End Sub

```

Commentaire du code

1 : cette routine contient un paramètre qui renseigne le genre de document à publiposter.
 Ici, il s'agira de « Attestations », d'autres cas seront évoqués plus loin [ici](#) et [ici](#)).

2-5 : on définit des variables, dont **Dim objWord As Word.Application** qui implique que l'on ajoute la bibliothèque **Microsoft Word xx.x Object Library** au projet (« xx.x » correspond à la version installée sur votre machine)

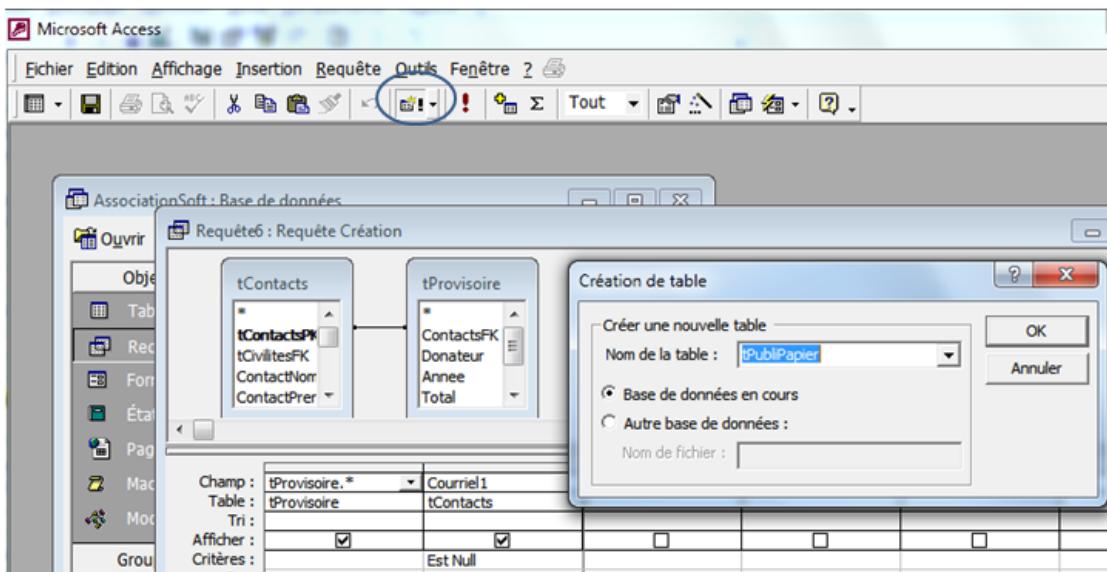


10-41 : on ajuste le contenu de certaines variables en fonction du paramètre.

13 : dans notre cas, le document modèle décrit **plus haut** a été logé dans le sous-répertoire « **ATTESTATIONS** » et s'appelle « **AttestationsModele.doc** ».

14 : le document issu du publipostage sera logé dans le sous-répertoire « **AEnvoyer** » et s'appellera « **AttestationsAEnvoyer.doc** ».

17-20 : on crée une table **tPubliPapier**, le SQL correspond à cette requête :



En clair, on prend dans **tProvisoire** tous les contacts qui n'ont pas d'adresse e-mail.

22-41 : ces instructions concernent d'autres valeurs du paramètre.

43-54 : le publipostage proprement dit :

45 : on interface Access avec Word ;

48-53 : on déclenche la procédure de publipostage dans Word, avec le document **ModeleDoc** et on loge le résultat à l'adresse **SavedDoc**.

58-59 : on referme Word et on libère la mémoire.

63 : on supprime la table **tPubliPapier** désormais inutile.

67 : on ouvre le document à publiposter : l'utilisateur n'a plus qu'à l'imprimer et à le mettre sous enveloppe.

Remarquez cette syntaxe « passe-partout »

 Shell "C:\WINDOWS\EXPLORER.EXE" & LeCheminDunFichier

Cela correspond à un double-clic sur le fichier quand vous êtes dans Explorer : le fichier s'ouvre avec le programme qui lui est associé.

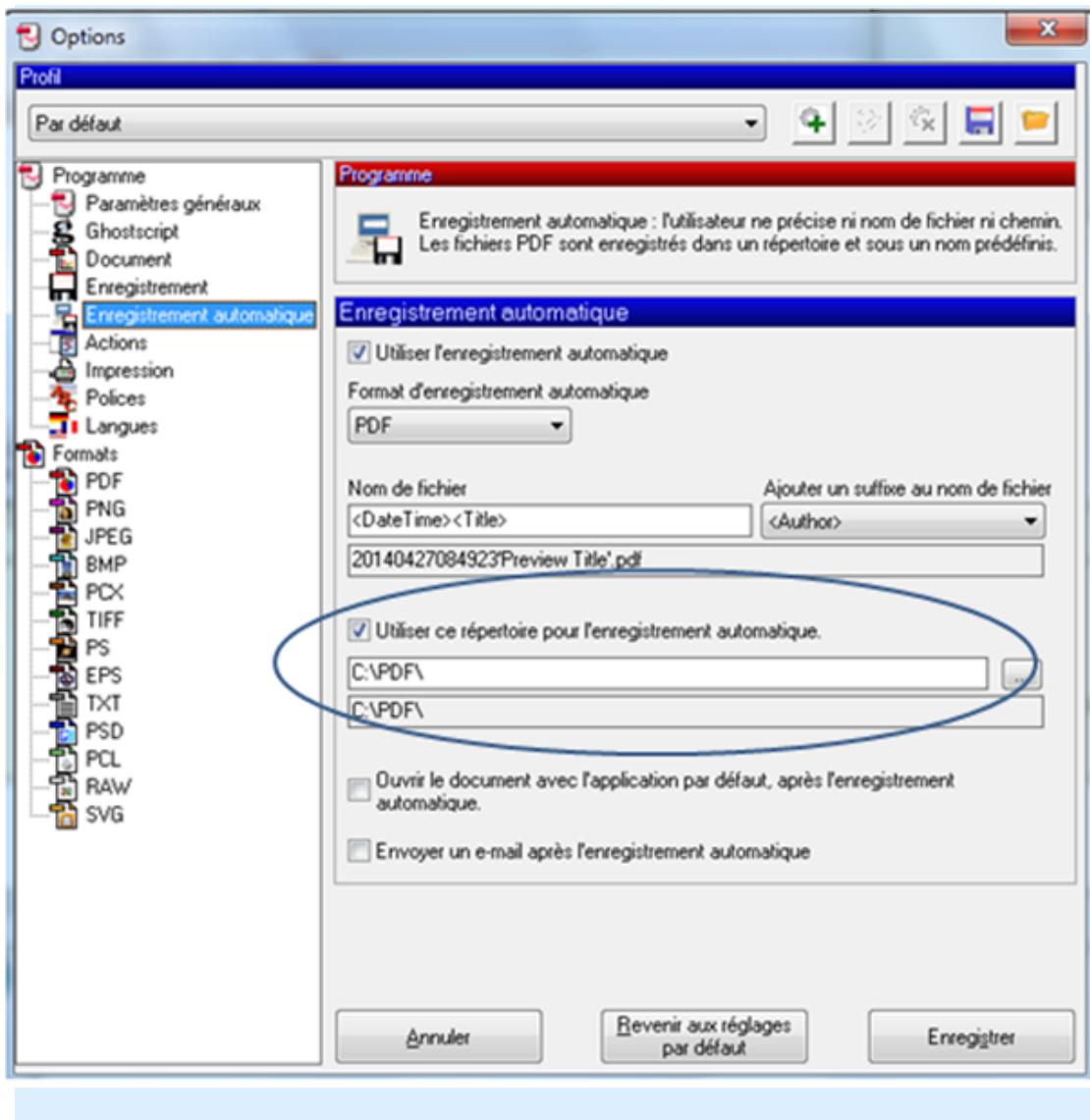
IV-F - Une procédure pour publiposter des lettres qui seront envoyées par e-mail

Pour utiliser le code suivant, vous devez disposer de PDFCreator, c'est un programme gratuit qui vous permet d'installer une imprimante qui, au lieu de produire du papier, crée un fichier .pdf.

 Chercher sur internet un endroit pour le télécharger.

Voici les paramètres qui ont été choisis pour l'installation : l'enregistrement automatique dans le répertoire C:\PDF\ (1)

Menu Imprimante > Options > Enregistrement automatique

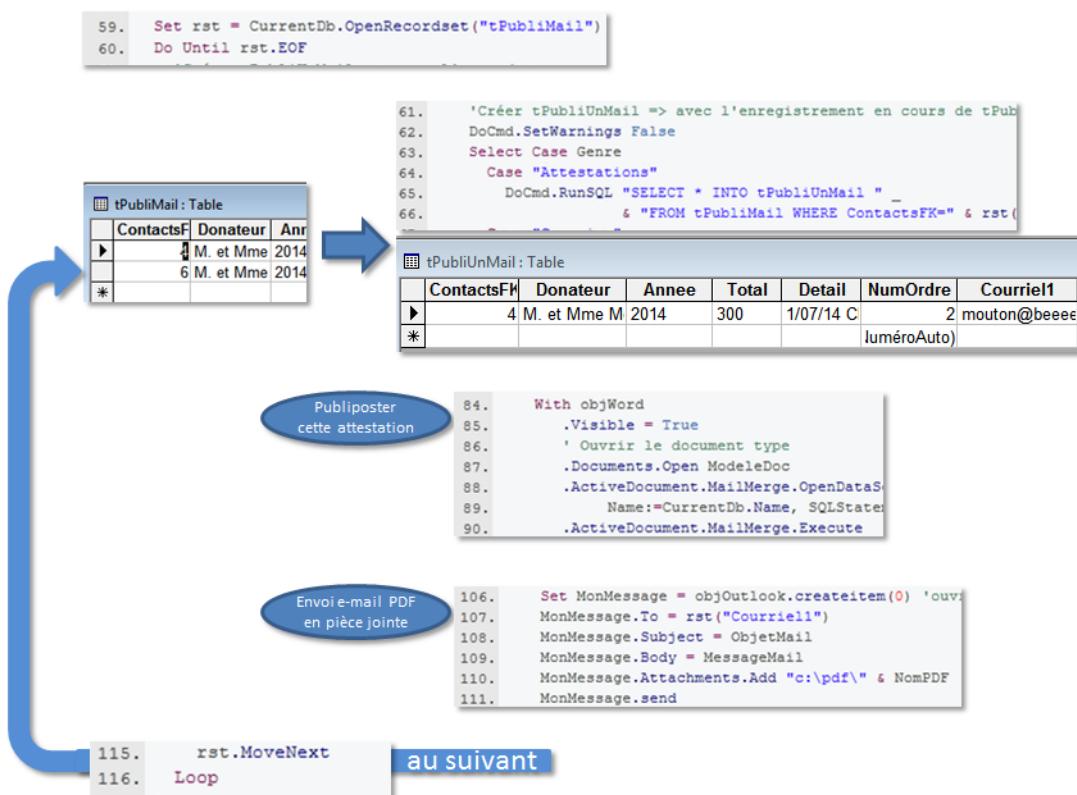


IV-F-1 - Les étapes

Préparer le terrain



Boucler pour publiposter un à un et envoyer par e-mail



Faire le ménage avant de partir

Rétablissement
imprimante papier

```
120. wsn.SetDefaultPrinter sImprDefaut
121. Set wsn = Nothing
122.
```

Refermer Word
et Outlook

```
124. objOutlook.Quit
125. Set objOutlook = Nothing
126. objWord.Quit
127. Set objWord = Nothing
128.
```

Supprimer les
tables inutiles

```
134. DoCmd.DeleteObject acTable, "tPubliMail"
135. DoCmd.DeleteObject acTable, "tPubliUnMail"
136.
```

IV-F-2 - Le code

```

1. Public Sub PubliMail(Genre As String)
2.   Dim objOutlook As Outlook.Application
3.   Dim MonMessage As Object
4.   Dim wsn As Object
5.   Dim rst As Recordset
6.   Dim sImprDefaut As String
7.   Dim oFSO As Scripting.FileSystemObject
8.   Dim oFld As Folder
9.   Dim oFl As File
10.  Dim objWord As Word.Application
11.  Dim NomDoc As String
12.  Dim ModeleDoc As String
13.  Dim NomPDF As String
14.  Dim ObjetMail As String
15.  Dim MessageMail As String
16.
17.  'Assigner l'objet Outlook et Word
18.  Set objOutlook = New Outlook.Application
19.  Set objWord = New Word.Application
20.
21.  'Changer l'imprimante par défaut
22.  '-----
23.  'Mémoriser l'imprimante par défaut
24.  sImprDefaut = ImprimanteParDefaut()
25.  'PDFCreator comme imprimante par défaut
26.  Set wsn = CreateObject("WScript.Network")
27.  wsn.SetDefaultPrinter "PDFCreator"
28.
29.  'Ajuster les paramètres en fonction du Genre
30.  '-----
31.  Select Case Genre
32.
33.    Case "Attestations"
34.      ModeleDoc = CurrentProject.Path & "\ATTESTATIONS\AttestationsModele.doc"
35.      NomPDF = "Attestation.pdf"
36.      ObjetMail = "Votre attestation fiscale"
37.      MessageMail = "Merci encore."
38.      'Créer la table tPubliMail
39.      DoCmd.SetWarnings False
40.      DoCmd.RunSQL "SELECT tProvisoire.*, tContacts.Courriell INTO tPubliMail " _
41.                    & "FROM tContacts INNER JOIN tProvisoire " _
42.                    & "ON tContacts.tContactsPK = tProvisoire.ContactsFK " _
43.                    & "WHERE Not tContacts.Courriell Is Null;"
```

```

44.     DoCmd.SetWarnings True
45.
46.     Case "Courrier"
47.         ModeleDoc = Forms!fEnvoiCourrier!txtCheminDoc
48.         NomPDF = "Courier.pdf"
49.         ObjetMail = Forms!fEnvoiCourrier!txtObjet
50.         MessageMail = Forms!fEnvoiCourrier!txtMessMail
51.         'Créer la table tPubliMail
52.         DoCmd.SetWarnings False
53.         DoCmd.RunSQL "SELECT * INTO tPubliMail FROM tCourrier WHERE tCourrier.[e-Mail]=True;"
54.         DoCmd.SetWarnings True
55.     End Select
56.
57.     'Lire chaque enregistrement de tPubliMail
58.     -----
59.     Set rst = CurrentDb.OpenRecordset("tPubliMail")
60.     Do Until rst.EOF
61.         'Créer tPubliUnMail => avec l'enregistrement en cours de tPubliMail
62.         DoCmd.SetWarnings False
63.         Select Case Genre
64.             Case "Attestations"
65.                 DoCmd.RunSQL "SELECT * INTO tPubliUnMail " _
66.                             & "FROM tPubliMail WHERE ContactsFK=" & rst("ContactsFK") & ";"
67.             Case "Courrier"
68.                 DoCmd.RunSQL "SELECT * INTO tPubliUnMail " _
69.                             & "FROM tPubliMail WHERE NomPrenom=""" & rst("NomPrenom") & """;"
70.         End Select
71.         DoCmd.SetWarnings True
72.
73.         'Créer le PDF de cette attestation
74.         -----
75.         'S'assurer que c:/pdf est vide
76.         Set oFSO = New Scripting.FileSystemObject
77.         Set oFld = oFSO.GetFolder("c:\pdf")
78.         For Each oFl In oFld.Files
79.             oFl.Delete
80.         Next oFl
81.
82.         'Publiposter
83.         -----
84.         With objWord
85.             .Visible = True
86.             ' Ouvrir le document type
87.             .Documents.Open ModeleDoc
88.             .ActiveDocument.MailMerge.OpenDataSource _
89.                 Name:=CurrentDb.Name, SQLStatement:="SELECT * FROM [tPubliUnMail]"
90.             .ActiveDocument.MailMerge.Execute
91.             .ActiveDocument.PrintOut
92.             .Documents.Close SaveChanges:=wdDoNotSaveChanges
93.         End With
94.         Sleep 2000 'pause de 2 s pour créer le document
95.
96.         'Renommer le fichier PDF
97.         -----
98.         For Each oFl In oFld.Files
99.             oFl.Name = NomPDF
100.        Next
101.
102.        'Expédier le mail avec le PDF en pièce jointe
103.        -----
104.
105.        'Composer le message
106.        Set MonMessage = objOutlook.createitem(0) 'ouvrir une structure de message
107.        MonMessage.To = rst("Courriell")
108.        MonMessage.Subject = ObjetMail
109.        MonMessage.Body = MessageMail
110.        MonMessage.Attachments.Add "c:\pdf\" & NomPDF
111.        MonMessage.send
112.        Sleep 2000 'pause de 2 s pour l'envoi
113.
114.        'Au suivant...
115.        rst.MoveNext

```

```

116. Loop
117. Sortie:
118.
119. 'Rétablir l'imprimante par défaut
120. wsn.SetDefaultPrinter sImprDefaut
121. Set wsn = Nothing
122.
123. 'Fermer Outlook et Word
124. objOutlook.Quit
125. Set objOutlook = Nothing
126. objWord.Quit
127. Set objWord = Nothing
128.
129. 'Libérer le recordset
130. rst.Close
131. Set rst = Nothing
132.
133. 'Supprimer les tables temporaires
134. DoCmd.DeleteObject acTable, "tPubliMail"
135. DoCmd.DeleteObject acTable, "tPubliUnMail"
136.
137. End Sub

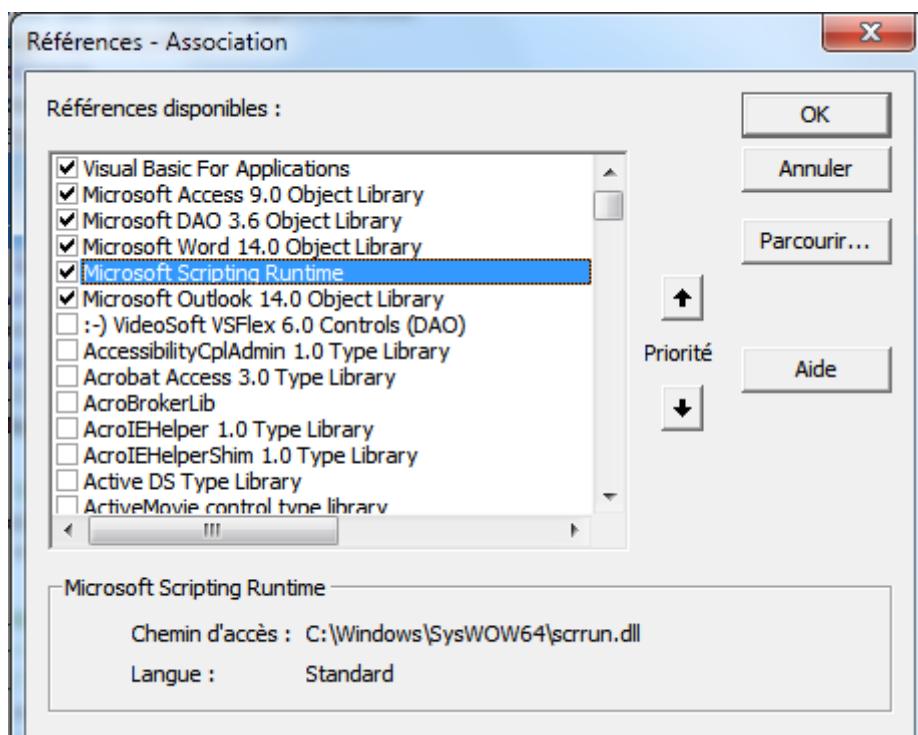
```

Commentaire du code

Cette routine contient un paramètre qui renseigne le genre de document à publiposter.
 Ici, il s'agira de « Attestations », d'autres cas seront évoqués plus loin ([ici](#) et [ici](#)).

2-15 : on définit des variables dont

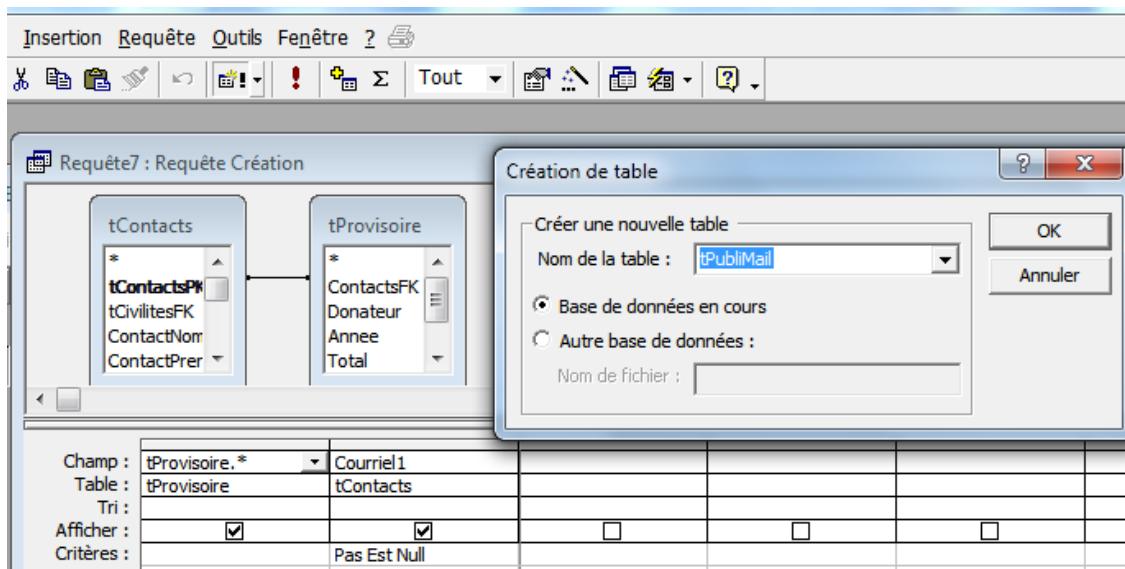
- Dim objOutlook As Outlook.Application qui implique que l'on ajoute la bibliothèque **Microsoft Outlook xx.x** au projet (« xx.x » correspond à la version installée sur votre machine) ;
- Dim oFSO As Scripting.FileSystemObject, Dim oFld As Folder et Dim oFl As File qui impliquent **Microsoft Scripting Runtime**.



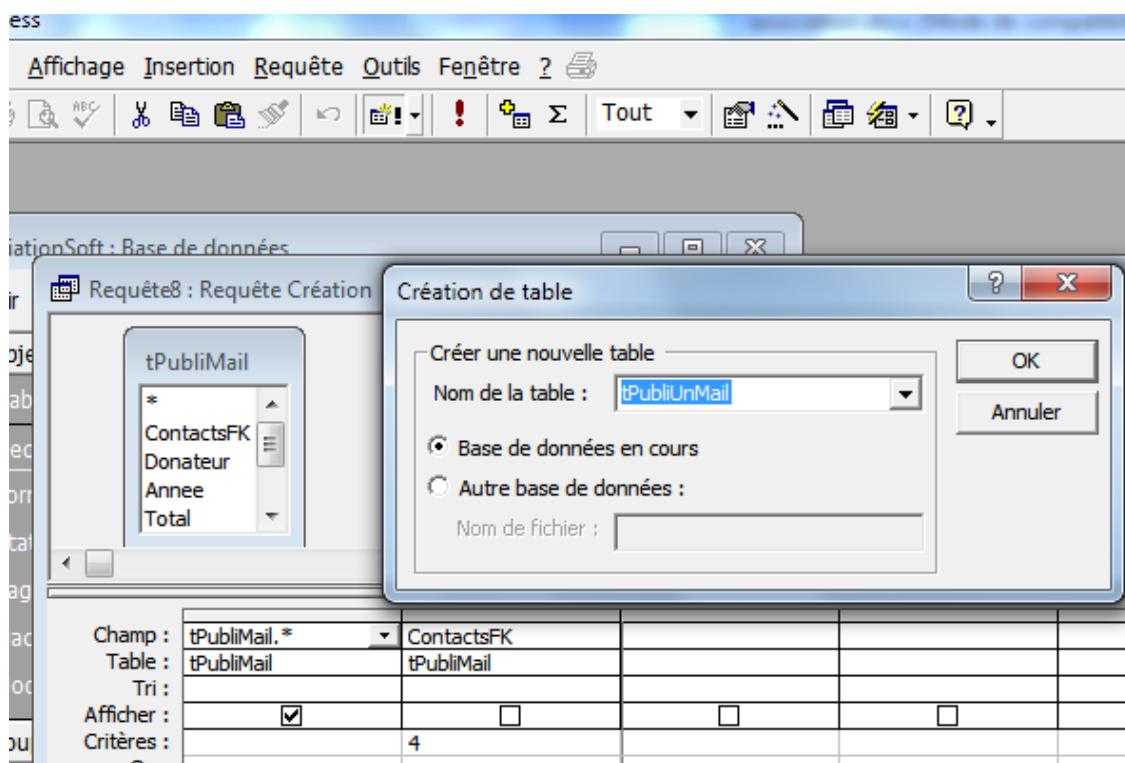
22-27 : on mémorise les coordonnées de l'imprimante actuellement par défaut (pour pouvoir la rétablir *in fine*) et on installe PDFCreator à la place.

24 : la fonction **ImprimanteParDefaut()** se trouve dans le module **mDivers**. (Désolé pour le manque d'explication, c'est un copier-coller de quelque part. Je n'ai pas tout compris, mais ça marche !)

40-43 : on crée une table **tPubliMail** qui contient tous les contacts à qui envoyer une attestation par e-mail. Le SQL généré correspond à ceci :



65-66 : dans la boucle de lecture des enregistrements de **tPubliMail**, on crée une table **tPubliUnMail** qui ne contient que le seul enregistrement en cours de lecture.

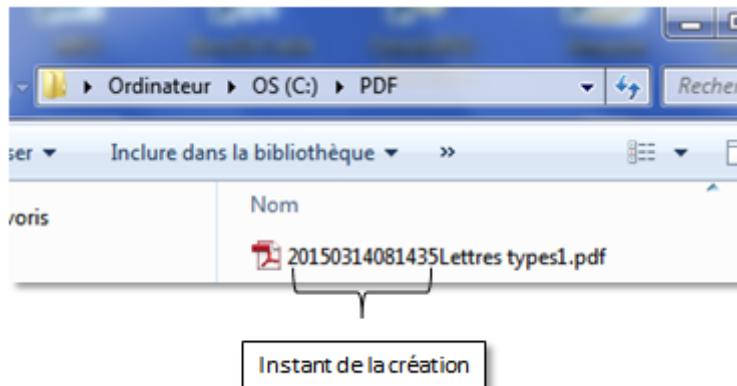


75-80 : on vidange le répertoire **c:\pdf**. Ce sera plus simple pour rebaptiser le fichier PDF que nous allons créer par publipostage.

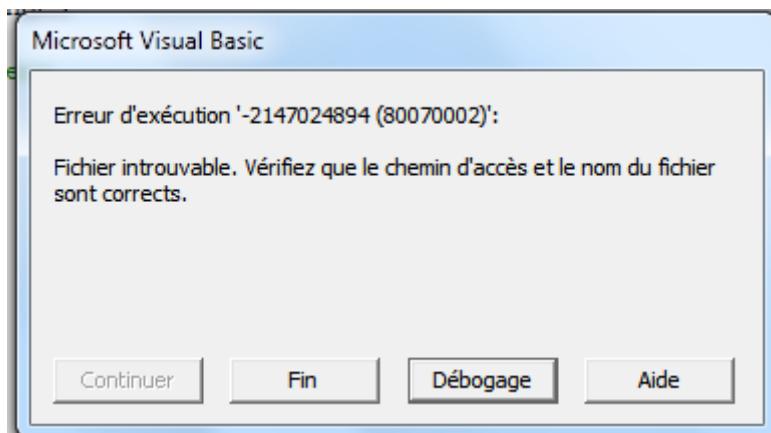


Il est donc impératif que le répertoire choisi pour loger les PDF générés par PDFCreator (dans l'exemple : c:\pdf) ne contienne aucun fichier que vous voudriez garder !

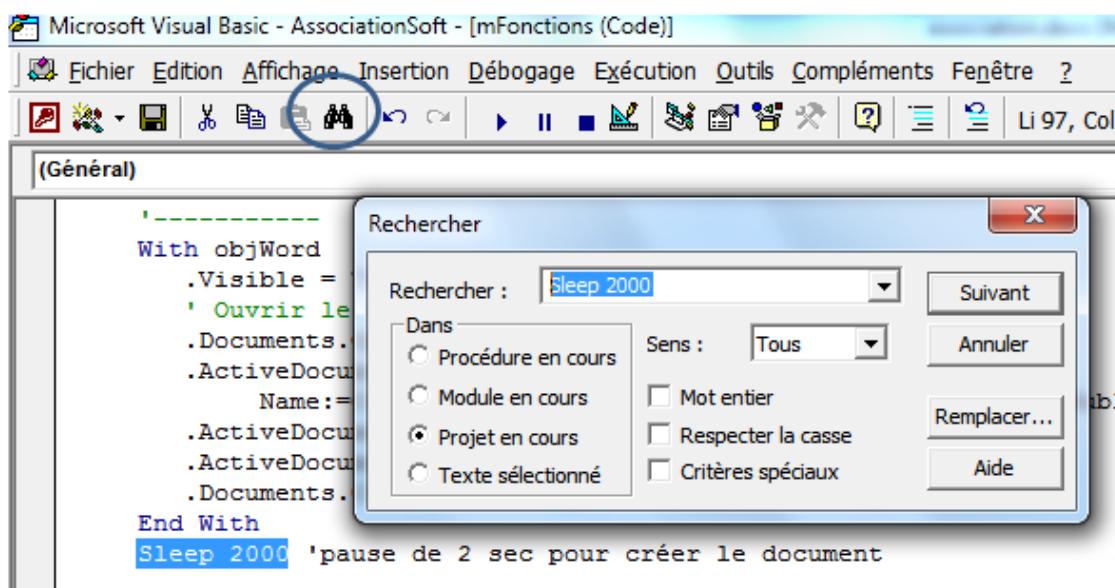
91 : le résultat du publipostage sera logé dans le répertoire c:\pdf et portera le nom que PDFCreator lui aura automatiquement donné. Quelque chose comme ceci :



94 : on fait une pause de deux secondes pour laisser le temps à Word de s'exécuter. En effet Access fonctionne de manière asynchrone, c'est-à-dire que dans le code une instruction n'attend pas que la précédente soit terminée. En l'occurrence, les instructions 98-100 qui vont rebaptiser le fichier PDF pourraient s'exécuter avant que le fichier PDF ne soit créé ! Ceci lèverait ce message d'erreur



Le délai nécessaire (ici 2 s) dépend de la vitesse de votre machine. Ajustez si nécessaire toutes les occurrences de Sleep dans tout le code du projet



98-100 : on rebaptise le fichier .pdf en lui donnant le nom « Attestation.pdf » (attribué à la variable **NomPDF** à l'instruction 35).

106-112 : on construit l'e-mail et on l'expédie avec Attestation.pdf en pièce jointe.

La suite du code n'appelle pas davantage de commentaires.

V - Archivage des attestations fiscales

Une copie des attestations doit être sauvegardée en vue d'un éventuel contrôle du fisc.

La procédure est une variante de ce que nous venons de décrire au chapitre précédent.

Nous publipostons sur papier pour l'ensemble des contacts concernés (qu'ils aient ou non une adresse e-mail) :

```

1. Private Sub btArchAttest_Click()
2.   'Créer tProvisoire
3.   DoCmd.SetWarnings False
4.   DoCmd.OpenQuery "rCreatProvisoire"
5.   DoCmd.RunSQL "ALTER TABLE tProvisoire ADD COLUMN NumOrdre COUNTER"
6.   DoCmd.SetWarnings True
7.   'Créer le document à archiver
8.   Call PubliPapier("ArchivAttest")
9.   'Supprimer la table tProvisoire
10.  DoCmd.DeleteObject acTable, "tProvisoire"
11. End Sub

```

'Créer le document à archiver
Call Publipapier("ArchivAttest")
'Copier dans la table tProvisoire



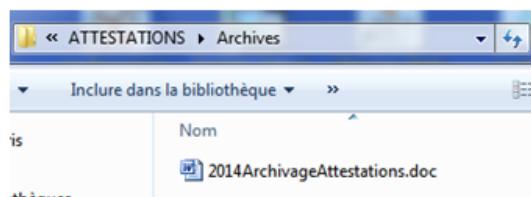
```

1. Public Sub Publipapier(Genre As String)

23.     Case "ArchivAttest"
24.         DoCmd.SetWarnings False
25.         DoCmd.RunSQL "SELECT tProvisoire.*, tContacts.Courriell INTO tPublipapier " - ①
26.                         & "FROM tContacts INNER JOIN tProvisoire "
27.                         & "ON tContacts.tContactsPK = tProvisoire.ContactsFK;"
28.         DoCmd.SetWarnings True
29.         ModeleDoc = CurrentProject.Path & "\ATTESTATIONS\AttestationsModele.doc"
30.         SavedDoc = CurrentProject.Path & "\ATTESTATIONS\Archives\" -
31.                         & DLookup("Annee", "tPublipapier") & "ArchivageAttestations.doc" ②
32.

```

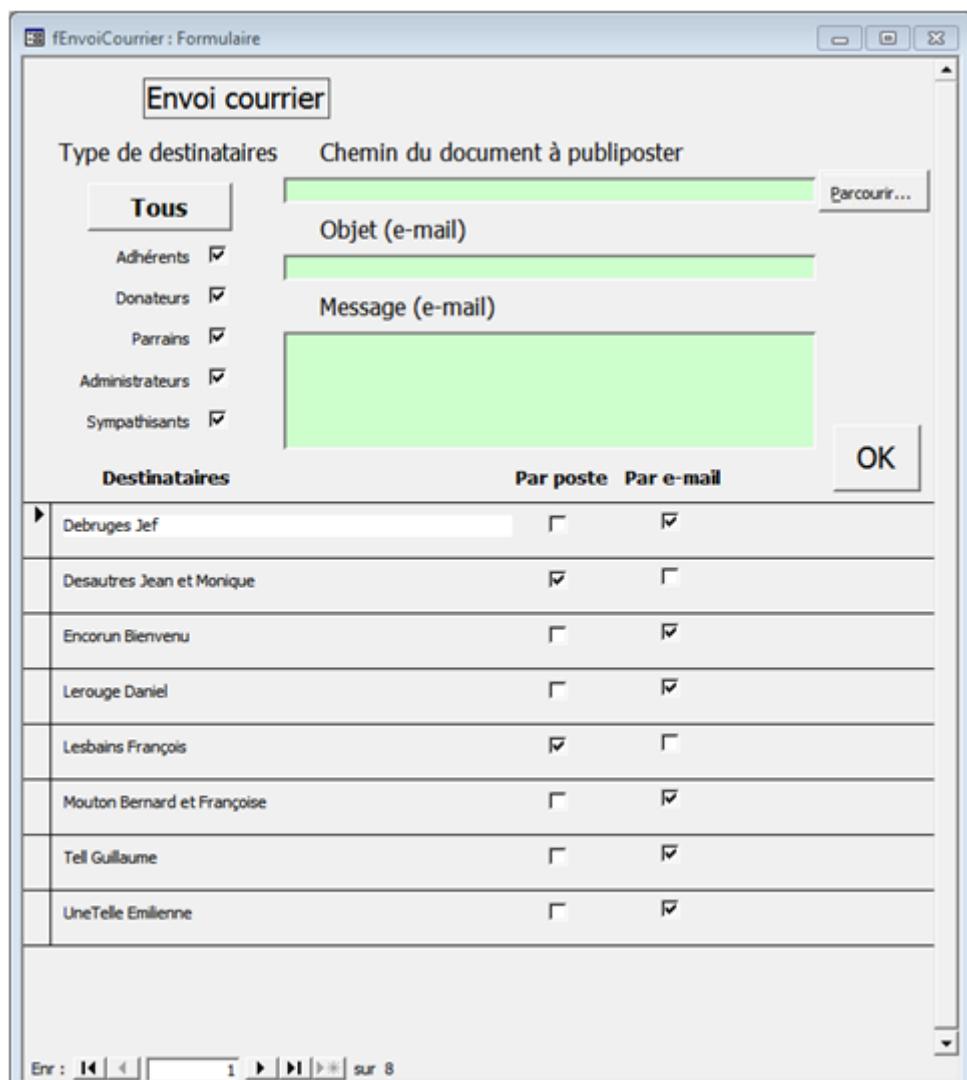
- 1 On prend tous les contacts qui ont eu une attestation
- 2 On sauvegarde le résultat de publipostage dans le sous-sous-répertoire « Archives »



VI - Envoi de courrier ciblé

VI-A - Le formulaire fEnvoiCourrier

À l'ouverture



- Tous les types de contacts sont sélectionnés comme destinataires. (Sans doublon pour les contacts qui ont plusieurs types d'engagements.)
- Pour éliminer des types de contacts, il suffit de décocher les cases non désirées

Type de destinataires

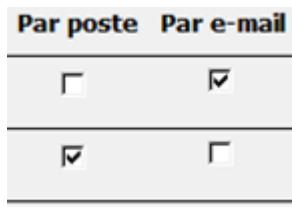
Adhérents	<input checked="" type="checkbox"/>
Donateurs	<input checked="" type="checkbox"/>
Parrains	<input checked="" type="checkbox"/>
Administrateurs	<input checked="" type="checkbox"/>
Sympathisants	<input checked="" type="checkbox"/>

- On peut aussi supprimer des enregistrements de manière manuelle, en les sélectionnant



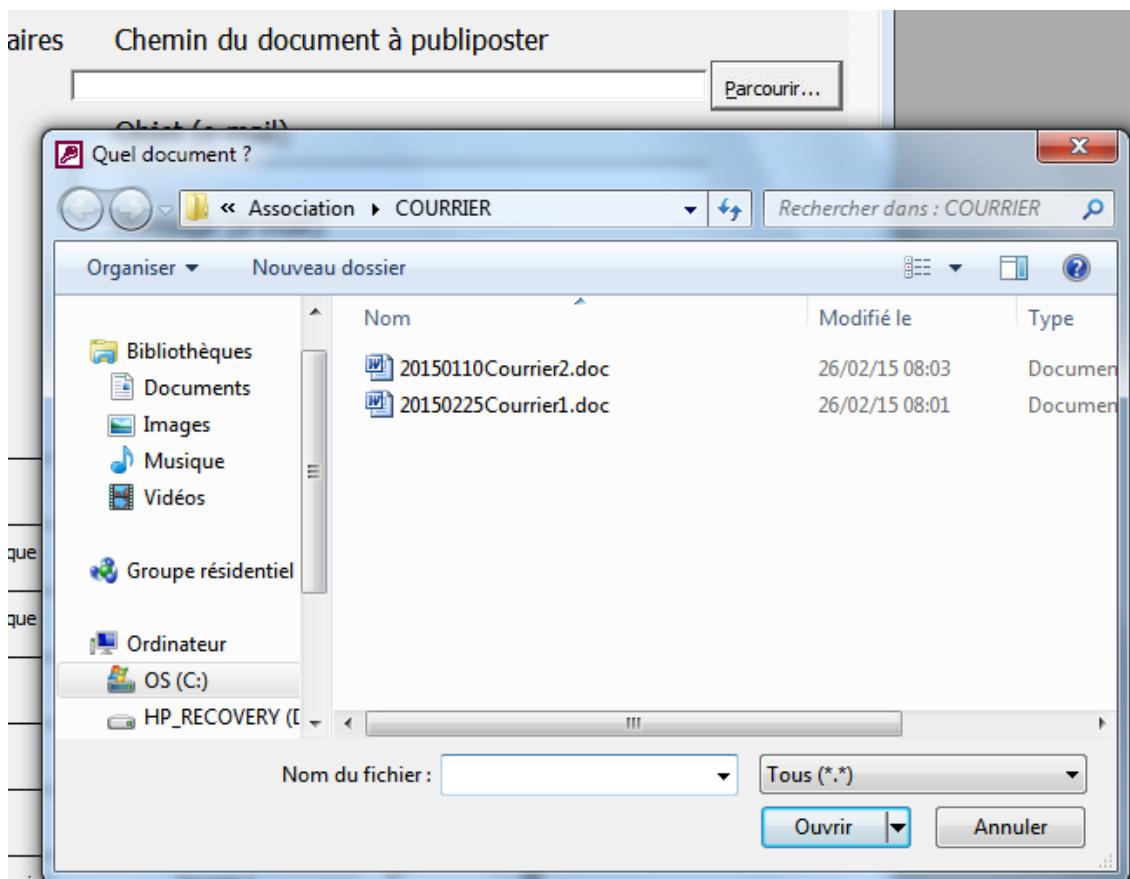
et en pressant la touche ou <Suppr>.

- Selon que le contact a donné une adresse courriel ou non, la case est cochée par défaut



(Il est possible de modifier ponctuellement la voie d'expédition pour autant qu'elle soit disponible pour ce contact.)

- Pour renseigner le chemin du document à publiposter, on peut :
 - soit l'introduire manuellement ;
 - soit cliquer le bouton Parcourir... qui ouvre une fenêtre de recherche de fichier.



- Cliquer le bouton OK déclenche le processus d'envoi.

VI-B - Le code

```

1. Option Compare Database
2. Option Explicit
3.
4. Private Sub Form_Open(Cancel As Integer)
5.     Me.RecordSource = ""
6.     DoCmd.SetWarnings False
7.     DoCmd.OpenQuery "rCreaCourrier"

```

```

8.  DoCmd.SetWarnings True
9.  Me.RecordSource = "tCourrier"
10. End Sub
11.
12. Private Sub Form_Unload(Cancel As Integer)
13.  Me.RecordSource = ""
14.  DoCmd.DeleteObject acTable, "tCourrier"
15. End Sub
16.
17. Private Sub BtChercher_Click()
18.  Me.txtCheminDoc = OuvrirUnFichier(Me.hwnd, "Quel document ?", 1, , ,
    CurrentProject.Path & "\COURRIER")
19. End Sub
20.
21. Private Sub btOK_Click()
22.  Dim oFSO As Scripting.FileSystemObject
23.  Me.Refresh
24.  'Vérifier complétude du formulaire
25.  If IsNull(Me.txtCheminDoc) + IsNull(Me.txtMessMail) + IsNull(Me.txtObjet) <> 0 Then
26.      MsgBox "Des champs obligatoires manquent"
27.      Exit Sub
28.  End If
29.  'Vérifier la présence du modèle
30.  Set oFSO = New Scripting.FileSystemObject
31.  If oFSO.FileExists(Me.txtCheminDoc) = False Then
32.      MsgBox "Le document à publiposter est introuvable", vbCritical
33.      Me.txtCheminDoc.SetFocus
34.      Exit Sub
35.  End If
36.  'Créer les lettres si demandées
37.  If DSum("Poste", "tCourrier") <> 0 Then Call Publipapier("Courrier")
38.  'Créer les e-mail si demandés
39.  If DSum("[e-mail]", "tCourrier") <> 0 Then Call PubliMail("Courrier")
40. End Sub
41.
42. Private Sub BtTous_Click()
43.  Dim ctl As Control
44.  For Each ctl In Me.Controls
45.      If ctl.Name Like "cc*" Then
46.          ctl = -1
47.      End If
48.  Next ctl
49.  Me.Repaint
50.  Call Form_Open(0)
51. End Sub
52.
53. Private Sub ccAdmi_AfterUpdate()
54.  Call Form_Open(0)
55. End Sub
56.
57. Private Sub ccDona_AfterUpdate()
58.  Call Form_Open(0)
59. End Sub
60.
61. Private Sub ccParr_AfterUpdate()
62.  Call Form_Open(0)
63. End Sub
64.
65. Private Sub ccSymp_AfterUpdate()
66.  Call Form_Open(0)
67. End Sub
68.
69. Private Sub ccAdhe_AfterUpdate()
70.  Call Form_Open(0)
71. End Sub
72.
73. Private Sub Poste_BeforeUpdate(Cancel As Integer)
74.
75.  If Me.Poste = True Then
76.      If IsNull(DLookup("Adresse1", "tCourrier", "NomPrenom=""" & Me.txtNomPrenom & """)) Then
77.          MsgBox "Ce contact n'a pas d'adresse postale", vbCritical
78.          Cancel = True

```

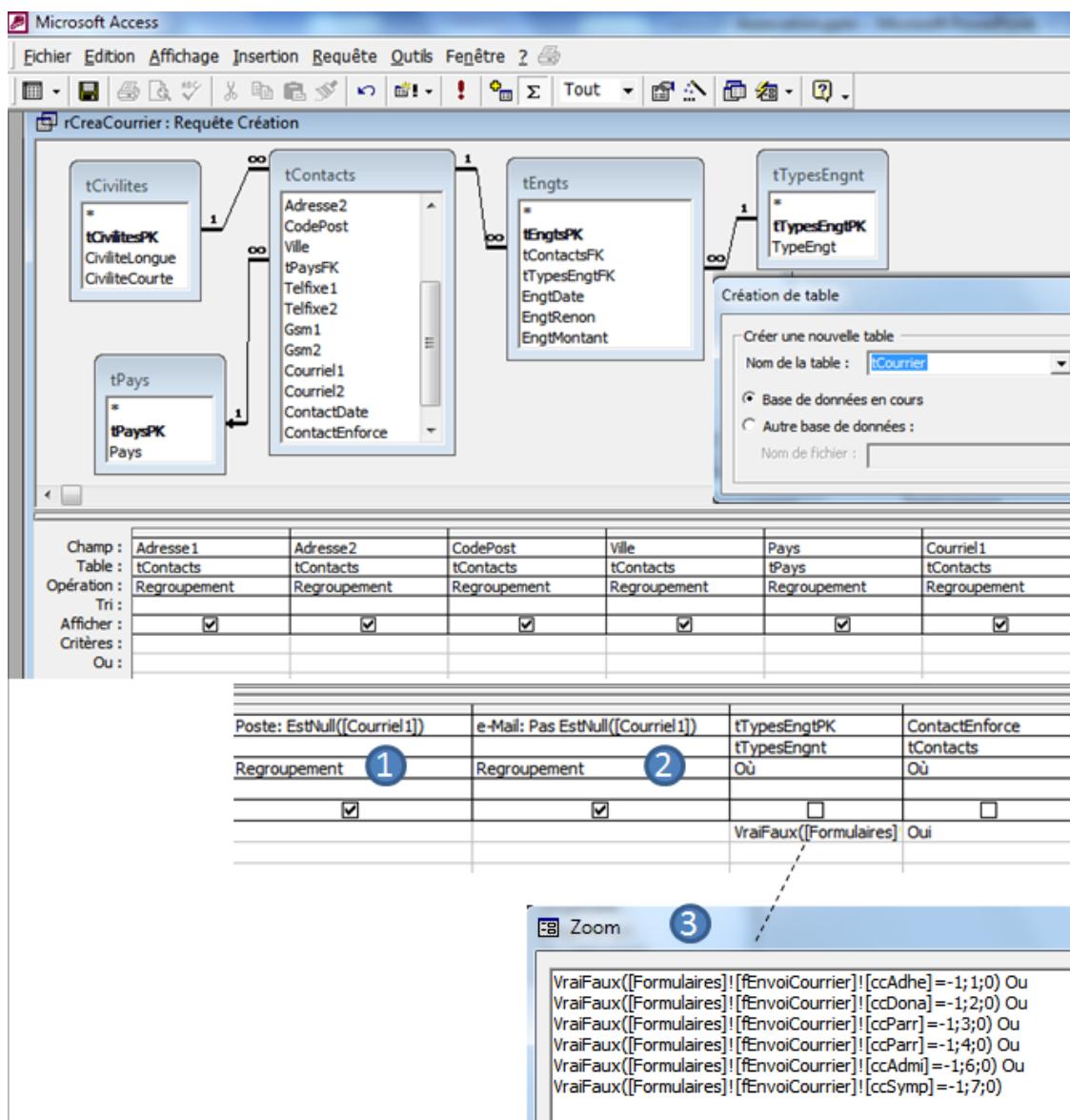
```

79.         Me.Poste.Undo
80.     End If
81. End If
82. End Sub
83.
84. Private Sub e_Mail_BeforeUpdate(Cancel As Integer)
85.     If Me.e_Mail = True Then
86.         If IsNull(DLookup("Courriel1", "tCourrier", "NomPrenom="" &
Me.txtNomPrenom & "")") Then
87.             MsgBox "Ce contact n'a pas d'adresse e-mail", vbCritical
88.             Cancel = True
89.             Me.e_Mail.Undo
90.         End If
91.     End If
92. End Sub

```

Commentaire du code

4-10 : à l'ouverture, on exécute la requête **rCreaCourrier** qui crée (provisoirement) une nouvelle table **tCourrier**.



1 la valeur de la colonne sera égale à -1 si le contact n'a pas d'adresse e-mail, 0 s'il en a une.

2

le contraire : la colonne sera égale à -1 si le contact a une adresse e-mail, 0 s'il n'en a pas.

3

seulement pour les tTypesEngrPK qui correspondent aux cases cochées dans le formulaire. (À l'ouverture, elles sont toutes cochées par défaut.)

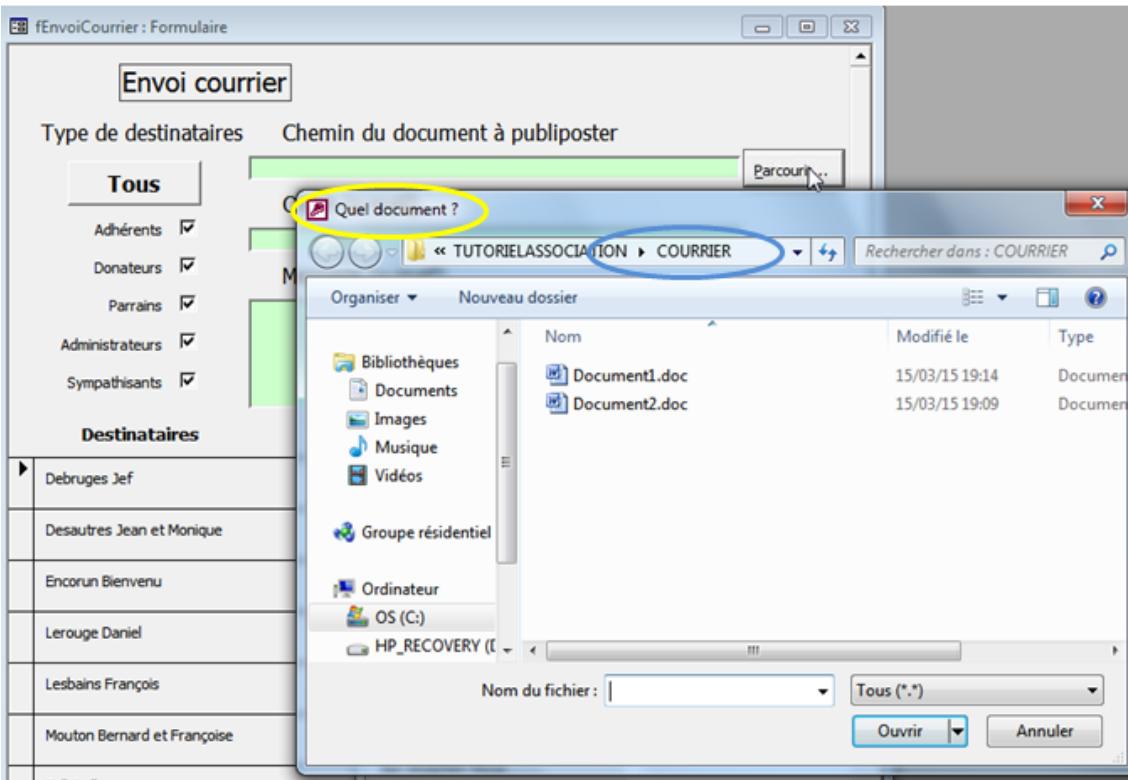
La table tCourrier ainsi créée :

Nom du champ	Type de données
CiviliteCourte	Texte
NomPrenom	Texte
Adresse1	Texte
Adresse2	Texte
CodePost	Texte
Ville	Texte
Pays	Texte
Courriel1	Texte
Poste	Numérique
e-Mail	Numérique

Cette table est affectée comme source au formulaire (instruction 9).

12-15 : la table sera supprimée lors de la fermeture du formulaire.

17-19 : un clic sur le bouton « Chercher » déclenche l'ouverture d'une boîte de dialogue pour le choix d'un fichier :



Le <http://access.developpez.com/faq/?page=CheminsRep - AffBoitDialogue> a été intégralement copié dans le module **mBoiteDialogue**.

Les paramètres de l'appel :

OuvrirUnFichier(Me.hwnd, "Quel document ?", 1, , CurrentProject.Path & "\COURRIER")

Le titre de la boîte

On récupère le chemin complet

On commence la recherche dans le sous-répertoire « Courrier ».
 (Dans l'exemple, l'application était logée dans le répertoire « TUTORIELASSOCIATION ».)

21-40 : lors d'un clic sur le bouton OK, on vérifie d'abord que l'utilisateur a complété les champs utiles pour publiposter.
 25 : la fonction **IsNull(Me.LeNomDuContrôle)** renverra « 0 » ou « -1 » selon que le champ a été complété ou non.
 Donc **IsNull(Me.txtCheminDoc) + IsNull(Me.txtMessMail) + IsNull(Me.txtObjet)** différent de zéro signifie qu'au moins un champ a la valeur *Null*.

37-39 : pour chacune des colonnes « Par poste » et « Par e-mail », pour autant qu'il y reste une case cochée, on appelle les procédures PubliPapier ou PubliMail déjà décrites plus haut ([ici](#) et [ici](#)).

42-51 : lors d'un clic sur le bouton « Tous », on parcourt tous les contrôles du formulaire et s'il s'agit d'un des filtres (leur nom commence par « cc »), on coche la case et on redéclenche le processus prévu à l'ouverture du formulaire.

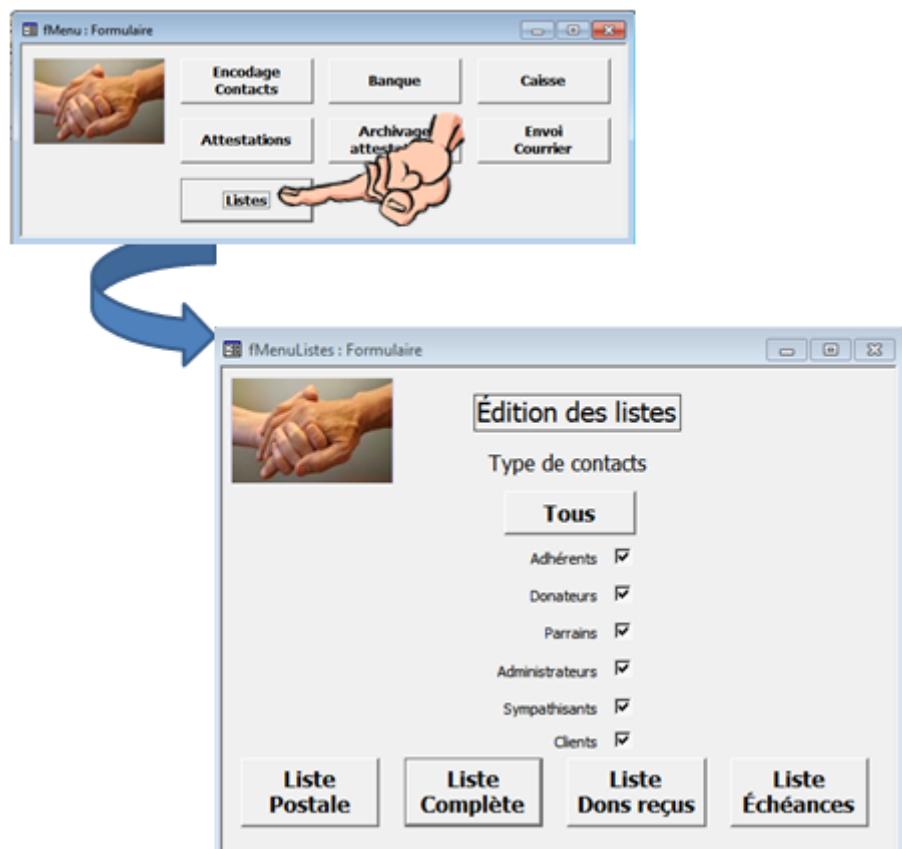
53-71 : lorsque l'état d'un des filtres est modifié, on redéclenche le processus prévu à l'ouverture du formulaire.

73-fin : avant d'accepter que l'utilisateur coche une case dans les colonnes « Par poste » et « Par email », on vérifie que le contact concerné dispose bien d'une telle adresse. Si ce n'est pas le cas, on affiche un message pour signaler le refus de l'action.

VII - Quelques états

VII-A - Un premier choix

Le formulaire fMenuListes permet de paramétriser les types de contacts concernés par l'état qui résultera du clic sur l'un des boutons.

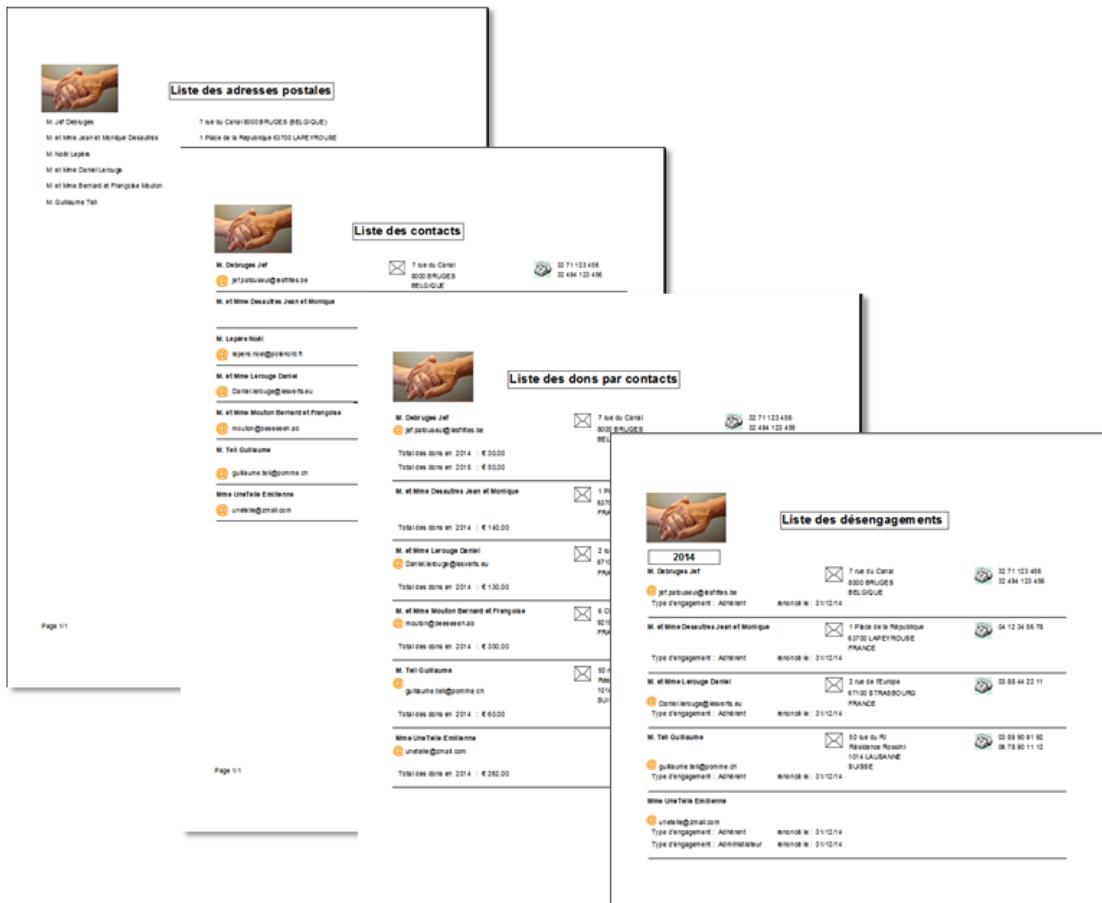


À l'origine toutes les cases sont cochées.

Un clic sur le bouton « Tous » recoche celles que l'utilisateur aurait décochées.

```
Private Sub BtTous_Click()
    Dim ctl As Control
    For Each ctl In Me.Controls
        If ctl.Name Like "cc*" Then
            ctl = -1
        End If
    Next ctl
End Sub
```

VII-B - Le code associé aux listes présentées ici



Peu de commentaire.

À l'ouverture, on aménage les adresses des images liées (même code que pour un formulaire)

```
Private Sub Report_Open(Cancel As Integer)
    Call AmnImages
End Sub
```

Dans l'événement « Sur formatage » de la section qui contient les adresses et téléphone, ce code

```
Private Sub Détail_Format(Cancel As Integer, FormatCount As Integer)
    Me.ImageCourriel.Visible = Not IsNull(Me.txtCourriel1)
    Me.ImageAdresse.Visible = Not IsNull(Me.txtAdresse1)
    Me.ImageTel.Visible = (IsNull(Me.txtTelfixel) + IsNull(Me.txtGsm1)) > -2
End Sub
```

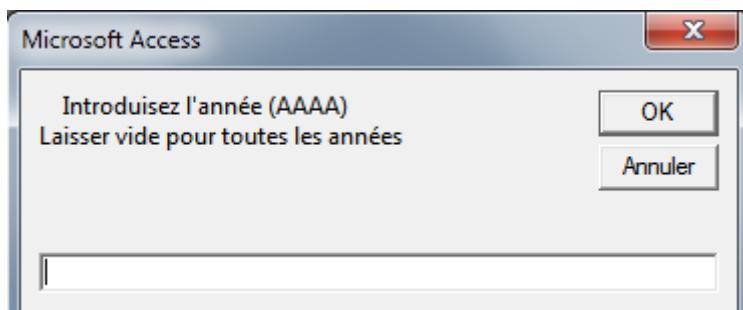
pour cacher l'icône si elle s'avère inutile.

Lorsqu'une année intervient dans l'état (par exemple dans « Liste des dons par contact »), le code est un peu plus étoffé :

- on invite l'utilisateur à saisir une année

```
'Choix de l'Année
Annee = InputBox("    Introduisez l'année (AAAA) " & vbCrLf & "Laisser vide pour toutes les
années")
```

cette fenêtre apparaît



- on construit alors à la volée la requête source de l'état, par exemple

```
'Aménager la source
sSelect = "SELECT Format([DonDate],"yyyy") AS Annee, Sum(rdonContact.DonMt) AS SommeDeDonMt,
"
" - & "[CiviliteCourte] & "" "" & [ContactNom] & "" "" & [ContactPrenom] AS NomPrenom, "
" - & "tContacts.Adresse1, tContacts.Adresse2, tContacts.CodePost,
tContacts.Ville, tPays.Pays, "
" - & "tContacts.Courriell, tContacts.Telfixe1, tContacts.Gsml,
tContacts.ContactNom, "
" - & "tContacts.ContactPrenom "
" - & " FROM tPays RIGHT JOIN ((tCivilites INNER JOIN (tContacts INNER JOIN
rdonContact "
" - & " ON tContacts.tContactsPK = rdonContact.tContactsFK) "
" - & "ON tCivilites.tCivilitesPK = tContacts.tCivilitesFK) INNER JOIN tEngts "
" - & "ON tContacts.tContactsPK = tEngts.tContactsFK) ON tPays.tPaysPK =
tContacts.tPaysFK "
" - & "WHERE (((tEngts.tTypesEngtFK) = IIf([Formulaires]![fMenuListes]![ccAdhe] = -1,
1, 0) "
" - & "Or (tEngts.tTypesEngtFK) = IIf([Formulaires]![fMenuListes]![ccDona] =
-1, 2, 0) "
" - & "Or (tEngts.tTypesEngtFK) = IIf([Formulaires]![fMenuListes]![ccParr] =
-1, 3, 0) "
" - & "Or (tEngts.tTypesEngtFK) = IIf([Formulaires]![fMenuListes]![ccParr] =
-1, 4, 0) "
" - & "Or (tEngts.tTypesEngtFK) = IIf([Formulaires]![fMenuListes]![ccAdmi] =
-1, 6, 0) "
" - & "Or (tEngts.tTypesEngtFK) = IIf([Formulaires]![fMenuListes]![
[ccSympa] = -1, 7, 0) "
" - & "Or (tEngts.tTypesEngtFK) = IIf([Formulaires]![fMenuListes]![ccCli] =
-1, 8, 0))) "
" - & "GROUP BY Format([DonDate],"yyyy"), [CiviliteCourte] & "" "" & [ContactNom] &
"" "" & [ContactPrenom], "
" - & "tContacts.Adresse1, tContacts.Adresse2, tContacts.CodePost,
tContacts.Ville, "
" - & "tPays.Pays, tContacts.Courriell, tContacts.Telfixe1,
tContacts.Gsml, "
" - & "tContacts.ContactNom, tContacts.ContactPrenom "

'Clause Having (dépend du choix de l'année)
If Len(Annee) = 4 Then 'L'utilisateur a choisi une année
    sHaving = "HAVING Format([DonDate],"yyyy") = "" & Annee & """
Else
    sHaving = "HAVING Format([DonDate],"yyyy") Like """*"""
End If
'Affecter la source
Me.RecordSource = sSelect & sHaving & ";"
```

Pour visualiser la représentation graphique d'un SQL écrit dans le code, vous pouvez utiliser l'interface graphique QBE (Query by Example).



Procédez comme ceci :

- ajouter cette instruction après la confection du SQL, ici, juste avant la dernière ligne

```
Debug.print sSelect & sHaving & ";"
```

- afficher l'état pour que le code se déclenche. Vous pouvez alors copier le SQL généré dans la fenêtre d'exécution et le coller dans une requête de test.

Pour vous familiariser avec cette technique, voyez ce tutoriel **Initiation - Débogage : requêtes écrites par VBA** de **Charles A (cafeine)** et singulièrement, le chapitre V.

VIII - Deux bases : une dorsale et une frontale



Pour approfondir le sujet, voyez le tutoriel de **Morgan Billy (Dolphy35)** **Comment utiliser une application en mode multiutilisateur**

Les tables permanentes et leurs relations, dans la dorsale **AssociationData.mdb** et tout le reste (requêtes, formulaires, états, macros et code) dans la frontale **AssociationSoft.mdb**.

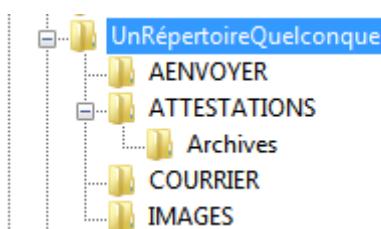
Les deux db sont logées dans le même répertoire.

Cette organisation permet de répartir la mise à jour des tables entre plusieurs utilisateurs. Par exemple l'un se charge de l'enregistrement des dons (fCaisse et fBanque) et l'autre de tout ce qui est administratif (fContacts, envoi des attestations...).

Chacun prend la main tour à tour. Quand l'un a terminé sa gestion, il envoie **AssociationData.mdb** via e-mail à l'autre et lui cède la main. Celui qui prend la main, remplace le fichier AssociationData.mdb qu'il avait sur son PC par celui qu'il vient de recevoir par e-mail. Il opère ses changements et recède la main. Et on est reparti pour un tour...

IX - Une arborescence particulière

Ceci pour faciliter la maintenance et alléger le code (utilisation de répertoires relatifs)



Les deux db, **AssociationSoft.mdb** et **AssociationData.mdb**, sont logées dans un répertoire quelconque (sur le disque dur, sur un réseau, sur une clé USB...).

Ce répertoire contient des sous-répertoires :

AENVOYER : qui recueillera le résultat du dernier publipostage à mettre sous enveloppe ;

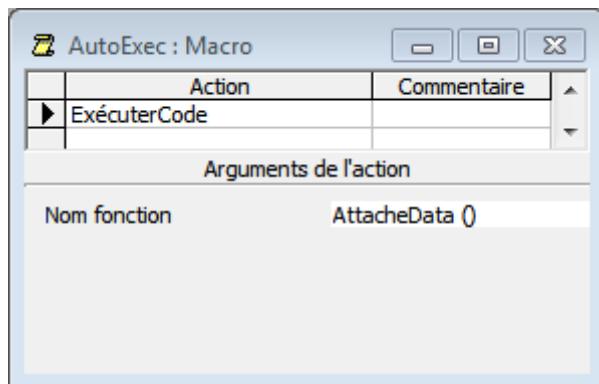
ATTENTATIONS : contient le document modèle pour le publipostage des attestations fiscales. Son sous-répertoire **Archives** contiendra la copie annuelle des attestations destinée au contrôle fiscal éventuel ;

COURRIER : les différents modèles de lettres à publiposter ;

IMAGES : les images liées aux formulaires et états.

IX-A - Adaptation automatique de la liaison dorsale/frontale, si changement de localisation

À chaque ouverture de la base, la macro « Autoexec » se déclenche



La fonction **AttacheData()** est logée dans le module **mAttacherTables**.

```

1. Option Compare Database
2. Option Explicit
3.
4. Public Function AttacheData()
5. On Error GoTo GestionErreurs
6. Dim db As Database
7. Dim tdfLoop As TableDef
8. Dim CheminSoft As String
9. Dim CheminData As String
10. CheminSoft = CurrentDb.Name
11. CheminData = Left(CheminSoft, Len(CheminSoft) - 8) & "data.mdb"
12. 'Vérifie que l'emplacement n'a pas changé depuis la fois précédente
13. If DLookup("Database", "MSysObjects", "not isnull(Database)") = CheminData Then Exit Function
    'le répertoire n'a pas été changé
14. 'Pour attacher les tables de data
15. Set db = OpenDatabase(CheminData)
16. For Each tdfLoop In db.TableDefs
17.     If Left(tdfLoop.Name, 4) <> "MSys" Then
18.         DoCmd.DeleteObject acTable, tdfLoop.Name
19.         DoCmd.TransferDatabase acLink, "Microsoft Access", CheminData, acTable,
    tdfLoop.Name, tdfLoop.Name
20.     End If
21. Next tdfLoop
22. MsgBox "L'attachement des tables a été actualisé"
23. db.Close
24. Set db = Nothing
25. Exit Function
26. GestionErreurs:
27. If Err.Number = 7874 Then 'la table n'est pas encore dans XXXSoft
28.     Resume Next
29. Else
30.     MsgBox "Erreur dans l'attachement des tables " & Err.Number & " " & Err.Description
31. End If
32. End Function

```

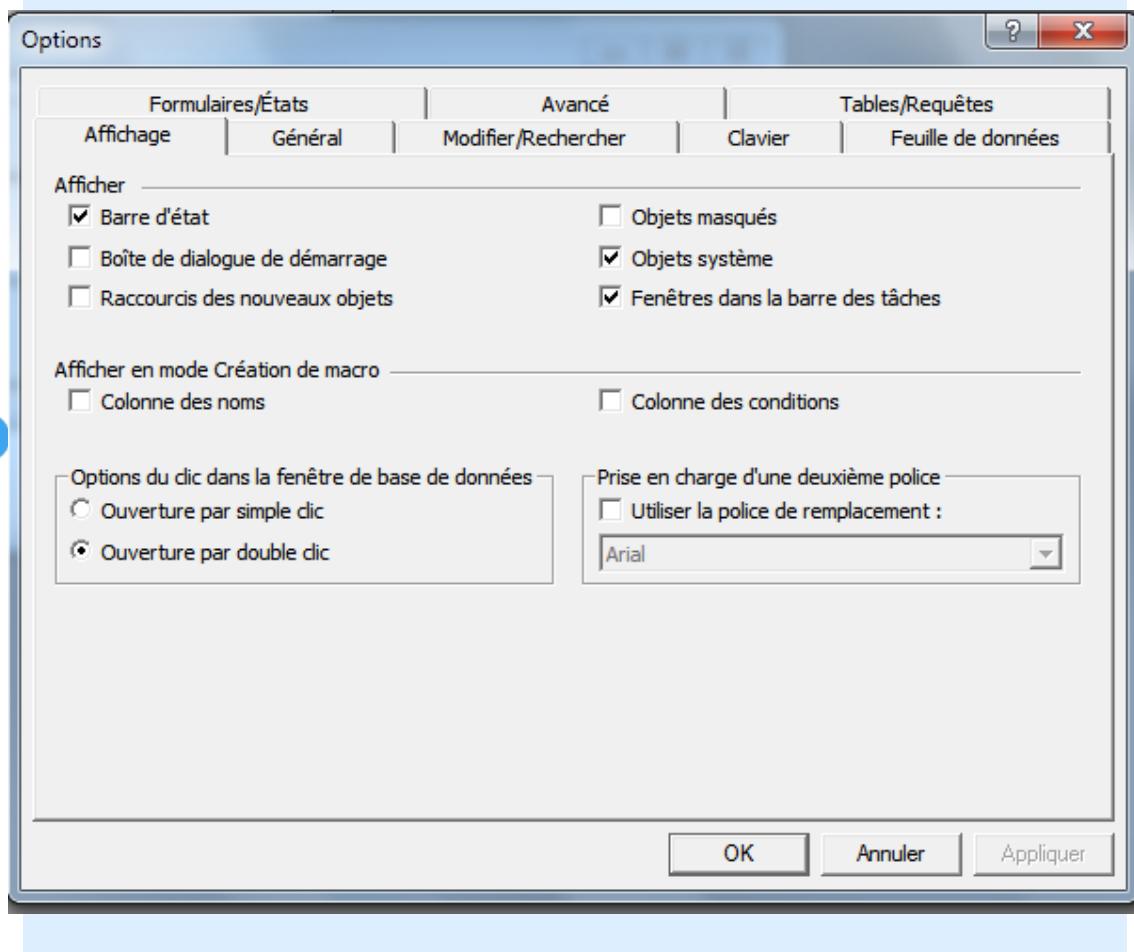
Explication du code

4 : AttacheData() ne renvoie pas de valeur, ce devrait donc être une **Sub**. On la définit néanmoins comme **Function** pour pouvoir la déclencher dans une macro (« AutoExec » en l'occurrence).

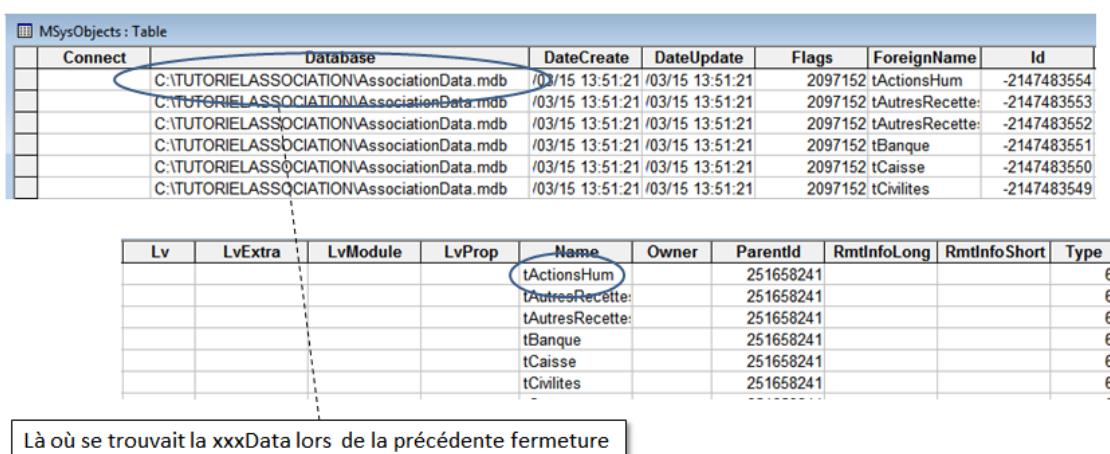
11 : on détermine quel devrait être le chemin de la xxxData.mdb, sachant qu'elle doit se trouver dans le même répertoire que la xxxSoft.mdb.

13 : on vérifie que ce chemin est bien celui enregistré dans la table système **MsysObjects**.

Pour afficher les tables système : Outils>Options...>Affichage cocher la case Objets système



La table **MSysObjects** contient une colonne « Database » qui renseigne l'adresse de la db qui contient chaque table liée



Connect	Database	DateCreate	DateUpdate	Flags	ForeignKey	Id
	C:\TUTORIELASSOCIATION\AssociationData.mdb	/03/15 13:51:21	/03/15 13:51:21	2097152	tActionsHum	-2147483554
	C:\TUTORIELASSOCIATION\AssociationData.mdb	/03/15 13:51:21	/03/15 13:51:21	2097152	tAutresRecette:	-2147483553
	C:\TUTORIELASSOCIATION\AssociationData.mdb	/03/15 13:51:21	/03/15 13:51:21	2097152	tAutresRecette:	-2147483552
	C:\TUTORIELASSOCIATION\AssociationData.mdb	/03/15 13:51:21	/03/15 13:51:21	2097152	tBanque	-2147483551
	C:\TUTORIELASSOCIATION\AssociationData.mdb	/03/15 13:51:21	/03/15 13:51:21	2097152	tCaisse	-2147483550
	C:\TUTORIELASSOCIATION\AssociationData.mdb	/03/15 13:51:21	/03/15 13:51:21	2097152	tCivilites	-2147483549

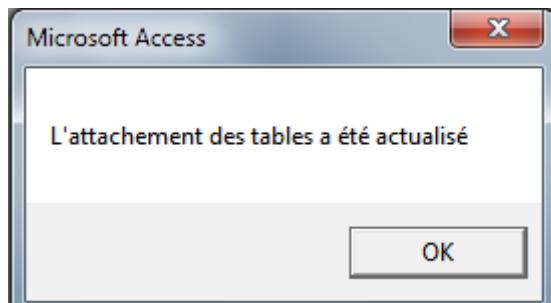
Lv	LvExtra	LvModule	LvProp	Name	Owner	ParentId	RmtInfoLong	RmtInfoShort	Type
				tActionsHum		251658241			6
				tAutresRecette:		251658241			6
				tAutresRecette:		251658241			6
				tBanque		251658241			6
				tCaisse		251658241			6
				tCivilites		251658241			6

Si on constate que l'adresse renseignée correspond à celle « calculée » en 11, on en conclut que les db se trouvent encore au même endroit que lors de l'exécution précédente. Tout va bien, on sort.

Par contre s'il y a différence, c'est que xxxSoft.mdb a changé d'emplacement et qu'il faut donc réactualiser les liaisons.

15-21 : réactualisation des liaisons. Pour chaque table (autre que les tables système) contenue dans xxxData.mdb, on supprime la table (liée) dans xxxSoft.mdb (18) et on établit la nouvelle liaison (19).

22 : on signale que le rafraîchissement des liaisons a été effectué :



X - Téléchargement

La db en version Access2000 se trouve ici.

XI - Remerciements

Ma gratitude à Joseph de l'association **vidica** qui a eu la patience de m'expliquer en détail le fonctionnement de son association.

Merci à **chrtope** pour ses remarques techniques très pertinentes.



Merci au talent d'Achille alias **f-leb** pour la relecture orthographique.

1 : Vous pouvez évidemment choisir une autre localisation et adapter la suite du code en conséquence. Seule précaution, le répertoire choisi ne doit rien contenir d'autre, car nous le vidangeons à chaque envoi.