

Découvrez une méthode pour partager vos contacts en réseau avec Outlook, mais sans Exchange Server

par Olivier Lebeau ([Heureux-oli sur DVP](#))

Date de publication : 20-11-2007

Dernière mise à jour :

Comment partager des contacts sur un réseau sans posséder Exchange.

- 1 - Introduction
- 2 - Création d'une base de données
- 3 - Récupérer les informations sur les contacts
- 4 - Connexion à la base de données
- 5 - Mettre les contacts à jour
- 6 - Comment lancer le code
- 7 - Remerciements

1 - Introduction

Le coût d'un serveur Exchange rend difficile le travail collaboratif pour les petites entreprises. Ce que je vais proposer est un partage de données concernant les contacts en utilisant une base de données Access.

Pourquoi Access ?

La réponse est assez simple, Access propose un type de fichier de données qui peut être ouvert par plusieurs utilisateurs en même temps.

Le principe que nous allons utiliser est simple, sur demande de l'utilisateur, nous allons faire une comparaison entre les données qui sont existantes dans une table et les données présentes dans les contacts.

Deux notions que je vais reprendre dans l'article : les contacts locaux (présents dans le carnet d'adresse) et les contacts distants (présents dans la base de données).

Nous allons ajouter les nouveaux contacts locaux dans la base de données et copier les nouveaux contacts distants dans le carnet d'adresse

La première étape sera la comparaison entre les données des contacts locaux et les contacts distants, seuls les contacts manquants dans la base de données seront ajoutés.

Le seconde étape fera le processus inverse, copier dans les contacts locaux les contacts distants qui auront été ajoutés par vos collaborateurs.

2 - Création d'une base de données

La première étape est la création d'une base de données qui va recevoir les informations des contacts du carnet d'adresses.

Par facilité, je vais placer cette base de données dans le "f:\Temp\)", mais il est possible de la mettre à n'importe quel endroit sur votre réseau. La seule chose à laquelle vous devez faire attention, ce sont les droits en écriture sur le répertoire où se trouvera la base de données.

Par souci de facilité, je vais utiliser un modèle de base de données toute faite, mais malheureusement, ce modèle ne fonctionne pas pour tous. Je vous conseille de faire une DB vide "Contacts.mdb" et d'y importer la table Contacts. Si vous ne possédez pas Access 2007, j'ai mis la liste des champs qui se trouvent dans la Base de données.

Dans un premier temps, j'ai tenté l'utilisation de la base de données disponible pour Access 2007 (Contacts.accdb). Par souci de portabilité, j'ai rétrogradé la base de données au format **mdb** Access 2003. Mais rien ne vous empêche de concevoir votre propre base de données.



Base de données
vide

Modèles en ligne



Biens



Contacts



Problèmes



Événements



Projets de marketing



Enseignants



Étudiants



Nouveautés dans Access 2007



La nouvelle version d'Access 2007 met à votre disposition

Inform
Forma

OI

La base de données contacts.

Options...

Du contenu de la base de données a été désactivé

Liste des contacts		Contacts	
	Nom du champ	Type de données	
	ID	NuméroAuto	
	Société	Texte	
	Nom	Texte	
	Prénom	Texte	
	Adresse de messagerie	Texte	
	Fonction	Texte	
	Téléphone professionnel	Texte	
	Téléphone personnel	Texte	
	Téléphone mobile	Texte	
	Numéro de télécopie	Texte	
	Adresse	Mémo	
	Ville	Texte	
	Département	Texte	
	Code postal	Texte	
	Pays/Région	Texte	
	Page Web	Lien hypertexte	
	Remarques	Mémo	
	Pièces jointes	Pièce jointe	

Général | Liste de choix

Taille du champ | Entier long

Les champs que nous avons dans notre table contacts sont :

- ID -> AutoNumber
- Société -> Texte
- Nom -> Texte
- Prénom -> Texte
- Adresse de messagerie -> Texte
- Fonction -> Texte

- Téléphone professionnel -> Texte
- Téléphone Personnel -> Texte
- Téléphone Mobile -> Texte
- Numéro de télécopie -> Texte
- Adresse -> Texte
- Ville -> Texte
- Département -> Texte
- Code postal -> Texte
- Pays -> Texte
- Page Web -> Texte

Cette table va nous servir pour stocker les données de notre carnet d'adresse.

Si la conception de votre propre base de données vous intéresse, le conseil que je peux vous donner est d'utiliser le nom des propriétés de ContactItems.

- LastName
- FirstName
- Email1address
- ...

3 - Récupérer les informations sur les contacts

Le principe est assez simple, nous allons parcourir les contacts et récupérer leurs informations.

Récupération des données

```
Public Sub ParcourirContact()  
Dim oCont As ContactItem  
Dim oFold As Folder  
Dim nM As NameSpace  
Dim olApp As Outlook.Application  
  
Set olApp = Outlook.Application  
Set nM = olApp.GetNamespace("MAPI")  
Set oFold = nM.GetDefaultFolder(olFolderContacts)  
For Each oCont In oFold.Items  
    Debug.Print oCont.LastName  
Next oCont  
  
End Sub
```

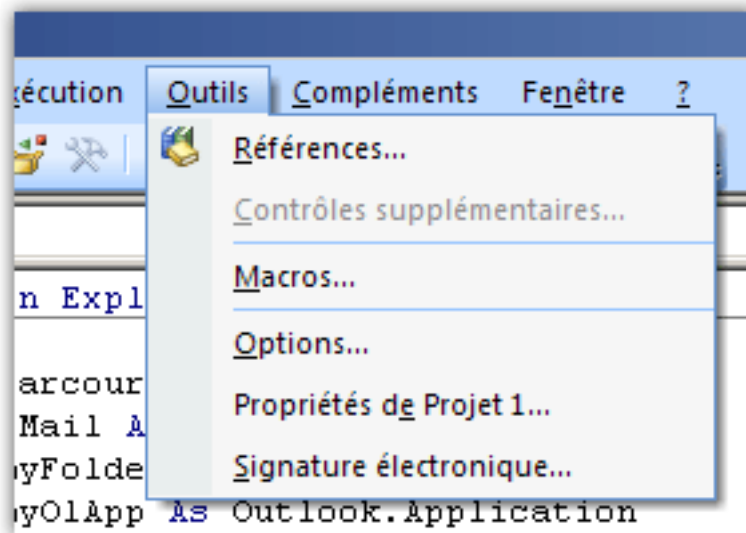
Dim oCont, oFol, nM, olApp sont utilisés pour déclarer les variables.

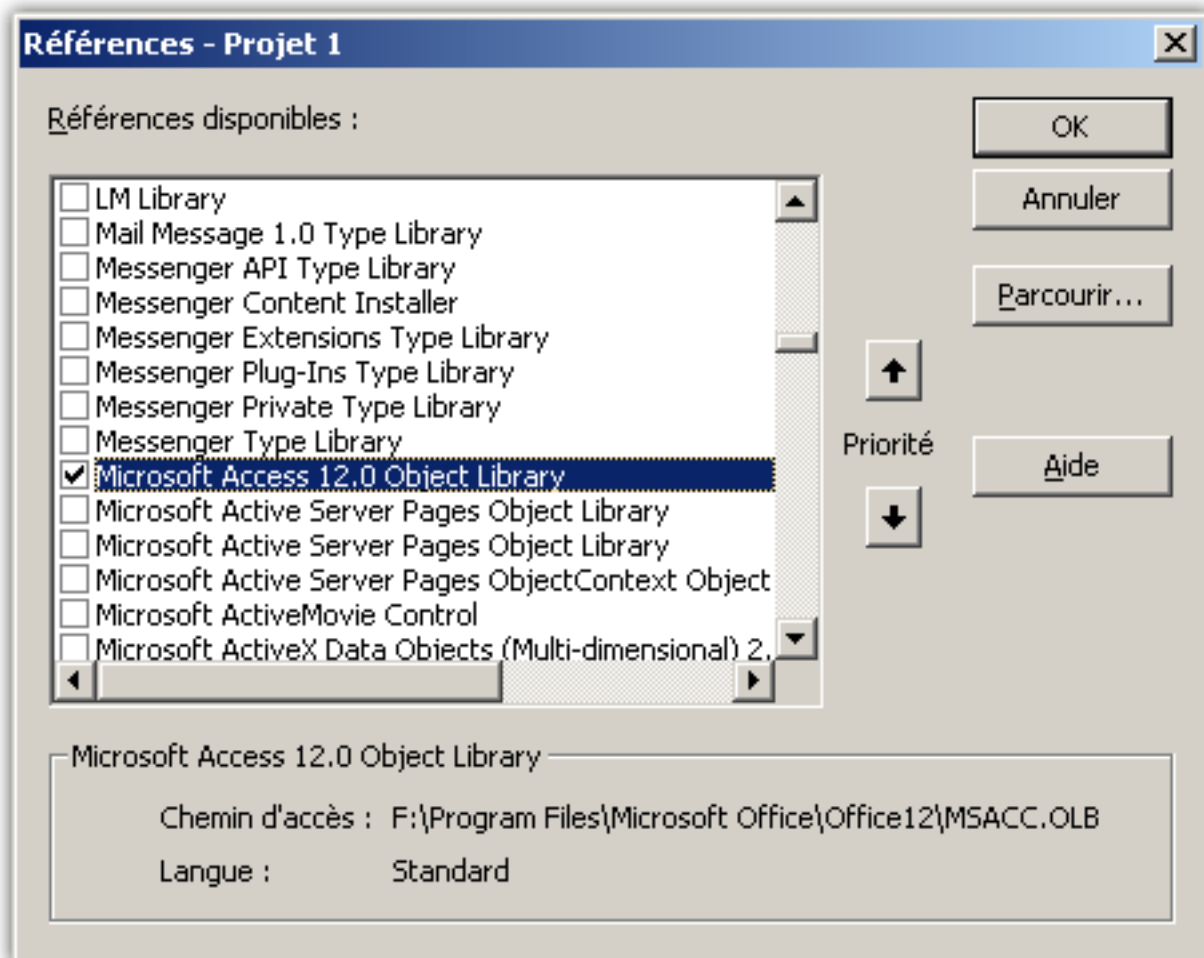
On va attribuer aux différents objets des valeurs par les méthodes Set, Set olAp, Set nM,...

Pour finalement parcourir les enregistrements du dossier contact et en extraire les données. Ce code n'étant qu'un exemple, j'ai volontairement mis un Debug.print sur une des propriétés du ContactItems.

4 - Connexion à la base de données

Pour nous connecter à la base de données, nous allons utiliser l'application Access. Il faut au préalable faire une référence à la bibliothèque Access dans votre projet et à la bibliothèque DAO.





Si l'exploration du DAO vous intéresse, je vous conseille de lire le tuto que Christophe Warin a écrit sur le sujet.

 **Le DAO**

```
Dim oAcc as Access.Application
```

Faire référence à la DB que nous avons créée.

```
Dim db as DAO.Database
Set db = OpenDatabase("F:\Temp>Contact.mdb")
```

Pour faire appel à la fonction, j'ai un peu modifié le code.

```
Public Sub ParcourirContact()
'*****
' Routine qui va parcourir les enregistrements présents dans le répertoire
' contacts et copier les enregistrements manquants dans la base de données
' Macro crée pour article DVP par Olivier Lebeau
'*****
Dim oCont As ContactItem
Dim oFold As Folder
```

```
Dim nM As NameSpace
Dim olApp As Outlook.Application
Dim i As Integer
Dim j As Integer

j = 1
' Affectation des objets
Set olApp = Outlook.Application
Set nM = olApp.GetNamespace("MAPI")
Set oFold = nM.GetDefaultFolder(olFolderContacts)

i = oFold.Items.Count
' Boucle pour parcourir les contacts locaux
For j = 1 To i
    ' Appel à la fonction AccesADB avec comme paramètre le contactItem
    AccesADB (oFold.Items(j))
Next j
End Sub
```

Voilà la fonction qui va écrire les données dans la DB.

Fonction de copie des contacts locaux vers les contacts distants

```
Public Function AccesADB(mycont As ContactItem)
'*****
' Fonction appelée pour envoyer vers la base de données les nouveaux
' contacts
' Fonction écrite pour article DVP par Olivier Lebeau
'*****
On Error Resume Next
Dim db As DAO.Database
Dim rs As DAO.Recordset
Dim sql As String
sql = "SELECT Contacts.*, Contacts.Nom, Contacts.[Prénom] "
sql = sql & " FROM Contacts "
sql = sql & " Where Contacts.Nom = "" & mycont.LastName
sql = sql & "" AND Contacts.[Prénom] = "" & mycont.FirstName & "";"
' Debug.Print sql
' Vous devez spécifier le chemin complet de votre base de données
Set db = OpenDatabase("f:\temp\contacts.mdb")
Set rs = db.OpenRecordset(sql)
' Debug.Print rs.RecordCount
'*****
' La liste des champs traités peut être augmentée en fonction de vos
' besoins. Par facilité, je n'ai volontairement mis que 3 champs
' Si vous rencontrez des problèmes avec les lignes Fields("xxxxx")
' je vous conseille d'utiliser l'index du champ Fields(2)
'*****

If rs.RecordCount = 0 Then
    rs.AddNew
    rs.Fields("Nom") = Nz(mycont.LastName, " ")
    rs.Fields("Prénom") = Nz(mycont.FirstName, " ")
    rs.Fields("Adresse de messagerie") = mycont.EmailAddress
    rs.Update
End If
'*****
' Libération des objets
'*****
rs.Close
db.Close
Set rs = Nothing
Set db = Nothing
End Function
```

La sentence SQL va vérifier que la ligne Nom Prénom n'existe pas, si c'est le cas, les données seront introduites dans la table, si les données existent, on continue pour le contact suivant.

Je n'ai inclus que trois champs dans le **Recordset**, mais ce n'est pas une limitation, vous pouvez en mettre plus et même créer votre propre DB.

5 - Mettre les contacts à jour

Sauvegarder les contacts sur le réseau n'a pas de sens si on ne peut faire la manipulation dans l'autre sens.

Pour mettre à jour le dossier contact, on va parcourir les enregistrements de la table et les comparer aux contacts. Si le contact est présent dans la liste, pas de problème, on passe à l'enregistrement suivant. Dès qu'un enregistrement n'est pas trouvé dans les contacts, il y est ajouté.

Mettre à jour les contacts locaux

```
Public Sub MettreAJourContact()  
'*****  
' Procédure pour récupérer les enregistrements présents dans la base de  
' données et les injecter dans le répertoire contact.  
'*****  
On Error Resume Next  
Dim oCont As ContactItem  
Dim oCo As ContactItem  
Dim oFold As Folder  
Dim nM As NameSpace  
Dim olApp As Outlook.Application  
Dim stFilt As String  
Dim rs As DAO.Recordset  
Dim db As DAO.Database  
  
'*****  
' Affectation des objets  
'*****  
  
Set db = OpenDatabase("F:\temp\contacts.mdb")  
Set rs = db.OpenRecordset("Select * From Contacts")  
Set olApp = Outlook.Application  
Set nM = olApp.GetNamespace("MAPI")  
Set oFold = nM.GetDefaultFolder(olFolderContacts)  
'*****  
' Boucle pour parcourir les enregistrements de la table  
'*****  
  
While Not rs.EOF  
'Filtre pour recherche des données déjà existantes dans les contacts locaux  
stFilt = "[FirstName] = "" & rs.Fields("Prénom")  
stFilt = stFilt & "" And [LastName] = "" & rs.Fields("Nom") & ""  
' Recherche avec filtre  
Set oCo = oFold.Items.Find(stFilt)  
' procédure décisionnelle pour copie des données  
If oCo = "Nothing" Then  
' Si pas de données, on les ajoute  
Set oCont = oFold.Items.Add  
oCont.FirstName = rs.Fields("Prénom")  
oCont.LastName = rs.Fields("Nom")  
oCont.EmaillAddress = rs.Fields("Adresse de messagerie")  
oCont.Save  
  
End If  
' Déplacement vers l'enregistrement suivant.  
rs.MoveNext  
Wend  
' Libération des objets  
rs.Close  
db.Close  
Set rs = Nothing  
Set db = Nothing  
End Sub
```

6 - Comment lancer le code

Le plus simple est de créer deux boutons dans la barre d'outils et de les faire pointer vers les deux Macros que nous venons d'écrire.

Clic droit sur une barre d'outils, Personnaliser, Commandes, Macros. Les deux macros que nous venons de faire apparaissent dans la liste. Il ne reste qu'un glisser déposer pour les insérer dans une barre d'outils.

7 - Remerciements

Un grand merci à toutes les personnes qui ont contribué et qui m'ont encouragé pour la rédaction de cet article, ans oublier N1bus.

Découvrez une méthode pour partager vos contacts en réseau avec Outlook,
mais sans Exchange Server par Olivier Lebeau ([Heureux-oli sur DVP](#))
