

Création et Utilisation des Bases SQL CE sous EVB 3.0

Introduction :

Pour commencer, on suppose que EVB et SQL CE 2.0 sont correctement installés.

Nous allons dans un premier temps utiliser les objets AdoXCE 3.1 pour créer la base, les enregistrements se feront par la suite avec les objets ADO 3.1, ainsi dans EVB il faut avoir sélectionné les bonnes références

-> Microsoft CE ADO Control 3.1

-> Microsoft CE ADO Ext. 3.1 for DDL

-> Microsoft CE SQL Server Control 2.0

N'activez pas le Microsoft CE ADO Control 3.0 puisque nous utilisons le 3.1 !

I) Création de la base à l'aide des objets ADOXE 3.1

Nous allons créer la base BaseExple, contenant deux tables :

First avec les colonnes :

-> Ref First en Integer et clé primaire (et donc unique)

-> Name en VarWChar

-> Nombre en Double

Second avec les colonnes :

-> RefSecond en Integer, clé primaire et Foreign Key sur First.ReffFirst

-> Check en Boolean

-> Bidule en Integer

Pour se faire nous avons besoin de définir les objets utilisés (vous trouverez la totalité du programme à la fin):

>>

```
Public sub CreateDatabase()
```

```
Dim oBase As ADOXCE.Catalog
```

```
Dim oTable As ADOXCE.Table
```

```
Dim oCol As ADOXCE.Column
```

```
Dim oKey As ADOXCE.Key
```

```
<<<
```

Puis nous créons la base à l'aide de l'objet oBase ... Remarquez au passage que nous testons les erreurs : à ce niveau, si erreur il y a, base déjà créée elle est (et on sort du programme)!

>>

```
On Error Resume Next
```

```
Set oBase = CreateObject("ADOXCE.Catalog.3.1")
```

```
oBase.Create "Provider=Microsoft.SQLSERVER.OLEDB.CE.2.0; data source=\My Documents\BaseExple.sdf"
```

```
If Err.Number <> 0 Then
```

```
    MsgBox "Base déjà"
```

```
    Err.Clear
```

```
    Set oBase = Nothing
```

```
Exit Sub
```

```
Else
```

```
    MsgBox "Première Connection, Création de la Base...", vbOKOnly, "Bienvenu"
```

```
End If
```

```
<<<
```

Nous allons créer la première table. La structure suivante est très importante, il faut bien la suivre et dans l'ordre (j'ai testé plusieurs syntaxes, seule celle-ci fonctionnait...)

>>

‘ On crée la table dans un premier temps, à l’aide d’un objet ADOXCE (ici oTable)

```
Set oTable = CreateObject("ADOXCE.Table.3.1")
```

```
oTable.Name = "First"
```

```
Set oTable.ParentCatalog = oBase ‘-> lie la Table First à la Base BaseExple via l’objet oBase
```

‘Il faut ensuite créer les colonnes ... On commence par renseigner l’objet oCol, puis on crée une première colonne à l’aide de la structure suivante :

```
Set oCol = CreateObject("ADOXCE.Column.3.1")
```

```
oCol.Name = "RefFirst"
```

```
oCol.Type = adInteger ‘= type Integer sous SQL CE 2.0
```

```
oTable.Columns.Append oCol ‘-> lie la colonne aux colonnes de la table First
```

```
oTable.ParentCatalog.Tables.Append oTable ‘-> la il enregistre vraiment la colonne
```

```
oBase.Tables.Refresh ‘-> Rafraichit la table dans oBase (avec la nouvelle colonne)
```

‘A partir de ce moment, on dispose d’une structure plus simple pour créer les colonnes ...

```
oBase.Tables("First").Columns.Append "Name", adVarChar, 100
```

```
oBase.Tables("First").Columns.Append "Nombrë", adDouble
```

<<<<

Voilà notre première table est créée, il s’agit maintenant de définir les clés ...

>>

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
```

```
oKey.Columns.Append "RefFirst" ‘colonne sur laquelle on place la clé
```

```
oKey.Type = adKeyPrimary ‘type de la clé (ici primaire)
```

```
oKey.Name = "FirstIDKey" ‘nom de la clé
```

```
oBase.Tables("First").Keys.Append oKey ‘-> lie la clé à la table First de la base
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
```

```
oKey.Columns.Append "RefFirst"
```

```
oKey.Type = adKeyUnique ‘Clé unique
```

```
oKey.Name = "FirstUniqueKey"
```

```
oBase.Tables("First").Keys.Append oKey
```

‘On « ferme » les objets clés, colonne et table pour pouvoir créer d’autres entrées (pas le oBase !!)

```
Set oKey = Nothing
```

```
Set oCol = Nothing
```

```
Set oTable = Nothing
```

<<<<

Et voilà, la table First avec ses trois colonnes et sa primary key est créée, on voit maintenant la table Second. Je remets l’ensemble des opérations sans explication, sauf pour la foreign key (commentaires en vert comme d’hab)!

Ps : On suppose que le code suivant est la suite de ce que l’on a fait (cf voir l’Annexe à la fin)

>>

```
Set oTable = CreateObject("ADOXCE.Table.3.1")
```

```
oTable.Name = "Second"
```

```
Set oTable.ParentCatalog = oBase
```

```
Set oCol = CreateObject("ADOXCE.Column.3.1")
```

```
oCol.Name = "RefSecond"
```

```
oCol.Type = adInteger
```

```
oTable.Columns.Append oCol
```

```
oTable.ParentCatalog.Tables.Append oTable
```

```
oBase.Tables.Refresh
```

```
oBase.Tables("Second ").Columns.Append "Check", adBoolean
oBase.Tables("Second ").Columns.Append "Bidule", adInteger
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
oKey.Columns.Append "RefSecond"
oKey.Type = adKeyPrimary
oKey.Name = "SecondIDKey"
oBase.Tables("Second").Keys.Append oKey
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
oKey.Columns.Append "RefSecond"
oKey.Type = adKeyUnique
oKey.Name = "SecondUniqueKey"
oBase.Tables("Second").Keys.Append oKey
```

‘ On crée maintenant la Foreign Key :

‘ Source : colonne RefSecond de la table Second -> Cible : colonne RefFirst de la table First

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
oKey.Type = adKeyForeign
oKey.Name = "FK_First_Second"
```

```
oKey.RelatedTable = "First"           ‘Nom de la table "Cible" de la FK
oKey.Columns.Append "RefSecond"      ‘Nom de la Colonne "Source", ie la colonne dans oTable (oTable = Second)
oKey.Columns("RefSecond").RelatedColumn = "RefFirst" ‘on lie la colonne source “RefSecond” à la cible “RefFirst”
oBase.Tables("Second").Keys.Append oKey ‘Nom de la Table Source, ici Second
```

```
Set oKey = Nothing
Set oCol = Nothing
Set oTable = Nothing
```

```
Set oBase = Nothing ‘on n’oublie pas de fermer l’objet oBase à la fin !
```

```
End Sub
```

```
<<<<
```

Vous l’aurez remarqué, l’ordre pour la FK n’est pas très heureux, il aurait été préférable de faire tous ce qui concerne la “Related Table” dans un premier temps, puis ce qui concerne la table Second... A tester ! (d’autres syntaxes existent appartement).

II) Utilisation de la Base

Dans un premier temps on initialise la base que l’on vient de créer (pour y avoir accès « Bonne appétit Bien sur »). Il faut appeler dans un module la variable :

Option Explicit

```
Public oTrucBase As ADOCE.Connection
```

Ensuite :

```
>>
```

```
Public Sub IniBase()
```

```
Set oTrucBase = CreateObject("ADOCE.Connection.3.1")
```

```
oTrucBase.ConnectionString = "provider=Microsoft.SQLServer.OLEDB.CE.2.0;data source=\My documents\ BaseExple.sdf;"
```

```
oTrucBase.Open
```

```
End Sub
```

```
<<<<
```

On peut maintenant appeler notre Base via oTrucbase (donnez lui le nom que vous voulez :Joel Robucchon :)

Passons aux choses sérieuses, créons nos premières entrées. L'instruction suivante initialise la table First. Notez bien la méthode Open, on en parle après.

```
>>
Public Sub ResetTable()

Dim rs As ADOCE.Recordset
Set rs = CreateObject("ADOCE.Recordset.3.1")

rs.Open "First", oTrucBase, adOpenDynamic, adLockOptimistic, adCmdTableDirect 'Voir commentaires après

If rs.EOF Then          'rs.EOF nous dit si l'on est à la fin du recordset (rs.BOF au début)
    rs.AddNew          'si oui, on crée une nouvelle ligne
End If

rs.Fields("RefFirst").Value = 1          'On change la valeur de la colonne RefFirst
rs.Fields("Name").Value = "Joel bien sur"
rs.Fields("Nombre").Value = 34.5678

rs.Update          'on update cette ligne du recordset, bien faire un update PAR LIGNE !

rs.Close
Set rs = Nothing

End Sub
<<<<
```

Et voilà, la table First contient la ligne suivante (1 ; Joel bien sur ; 34.5678), vous pouvez vérifier avec isqltrucmanchinchose sur le PDA. Si vous ne faites que lire le recordset ne faites pas de Update :Robuchon :

Bon maintenant quelques commentaires s'imposent en ce qui concerne la méthode Open du Recordset.

La syntaxe est la suivante :

```
oBjet.Open sSqlCommand, adOpenDynamic, adLockOptimistic [,adCmdTableDirect]
```

L'objet étant un Recordset, je vous passe les détails concernant adOpenDynamic et adLockOptimistic (allez voir la doc!).

Par contre en ce qui concerne le reste, deux solutions s'offrent à vous :

- ➔ Si vous voulez modifier les enregistrements d'une table !
sSqlCommand = "<Nom de la table>"
précisez adCmdTableDirect à la fin de l'Open.
- ➔ Si vous voulez juste avoir accès aux données d'une table (pas de modification)
Soit vous utilisez la méthode précédente, soit :
sSqlCommand = "<Instruction SQL>", par exemple "Select * From Second "
ne préciser pas le adCmdTableDirect

Voilà c'est tout, ENJOY !

(Ps : bien sur j'ai vérifié que tous fonctionnait sous EVB, et ça fonctionne ! non mais ;-)

ANNEXE : Programme en entier (faites appel à la fonction CreateDateBase)

Dans un module :

Option Explicit

Public oTrucBase As ADOCE.Connection

```
Public sub CreateDatabase()
```

```
Dim oBase As ADOXCE.Catalog
```

```
Dim oTable As ADOXCE.Table
```

```
Dim oCol As ADOXCE.Column
```

```
Dim oKey As ADOXCE.Key
```

```
On Error Resume Next
```

```
Set oBase = CreateObject("ADOXCE.Catalog.3.1")
```

```
oBase.Create "Provider=Microsoft.SQLSERVER.OLEDB.CE.2.0; data source=\My Documents\BaseExple.sdf"
```

```
If Err.Number <> 0 Then
```

```
    MsgBox "Base déjà"
```

```
    Err.Clear
```

```
    Set oBase = Nothing
```

```
Exit Sub
```

```
Else
```

```
    MsgBox "Première Connection, Création de la Base...", vbOKOnly, "Bienvenu"
```

```
End If
```

```
Set oTable = CreateObject("ADOXCE.Table.3.1")
```

```
oTable.Name = "First"
```

```
Set oTable.ParentCatalog = oBase
```

```
Set oCol = CreateObject("ADOXCE.Column.3.1")
```

```
oCol.Name = "RefFirst"
```

```
oCol.Type = adInteger
```

```
oTable.Columns.Append oCol
```

```
oTable.ParentCatalog.Tables.Append oTable
```

```
oBase.Tables.Refresh
```

```
oBase.Tables("First").Columns.Append "Name", adVarChar, 100
```

```
oBase.Tables("First").Columns.Append "Nombrë", adDouble
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
```

```
oKey.Columns.Append "RefFirst"
```

```
oKey.Type = adKeyPrimary
```

```
oKey.Name = "FirstIDKey"
```

```
oBase.Tables("First").Keys.Append oKey
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
```

```
oKey.Columns.Append "RefFirst"
```

```
oKey.Type = adKeyUnique
```

```
oKey.Name = "FirstUniqueKey"
```

```
oBase.Tables("First").Keys.Append oKey
```

```
Set oKey = Nothing
```

```
Set oCol = Nothing
```

```
Set oTable = Nothing
```

```
Set oTable = CreateObject("ADOXCE.Table.3.1")
oTable.Name = "Second"
Set oTable.ParentCatalog = oBase

Set oCol = CreateObject("ADOXCE.Column.3.1")
oCol.Name = "RefSecond"
oCol.Type = adInteger
oTable.Columns.Append oCol
oTable.ParentCatalog.Tables.Append oTable

oBase.Tables.Refresh

oBase.Tables("Second ").Columns.Append "Check", adBoolean
oBase.Tables("Second ").Columns.Append "Bidule", adInteger
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
oKey.Columns.Append "RefSecond"
oKey.Type = adKeyPrimary
oKey.Name = "SecondIDKey"
oBase.Tables("Second").Keys.Append oKey
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
oKey.Columns.Append "RefSecond"
oKey.Type = adKeyUnique
oKey.Name = "SecondUniqueKey"
oBase.Tables("Second").Keys.Append oKey
```

```
Set oKey = CreateObject("ADOXCE.Key.3.1")
oKey.Type = adKeyForeign
oKey.Name = "FK_First_Second"

oKey.RelatedTable = "First"
oKey.Columns.Append "RefSecond"
oKey.Columns("RefSecond").RelatedColumn = "RefFirst"
oBase.Tables("Second").Keys.Append oKey
```

```
Set oKey = Nothing
Set oCol = Nothing
Set oTable = Nothing
Set oBase = Nothing
```

```
IniBase
ResetTable
```

```
End Sub
```

```
Public Sub IniBase()
```

```
Set oTrucBase = CreateObject("ADOCE.Connection.3.1")
oTrucBase.ConnectionString = "provider=Microsoft.SQLServer.OLEDB.CE.2.0;data source=\My documents\ BaseExple.sdf;"
oTrucBase.Open
```

```
End Sub
```

```
Public Sub ResetTable()
```

```
Dim rs As ADOCE.Recordset
```

```
Set rs = CreateObject("ADOCE.Recordset.3.1")
```

```
rs.Open "First", oTrucBase, adOpenDynamic, adLockOptimistic, adCmdTableDirect
```

```
If rs.EOF Then
```

```
    rs.AddNew
```

```
End If
```

```
rs.Fields("RefFirst").Value = 1
```

```
rs.Fields("Name").Value = "Joel bien sur"
```

```
rs.Fields("Nombre").Value = 34.5678
```

```
rs.Update
```

```
rs.Close
```

```
Set rs = Nothing
```

```
End Sub
```