



Extrait du Environnement iSeries

<https://xdocs400.com/spip.php?article284>

Importer à partir d'Excel des données ISERIES

- Les articles -



Date de mise en ligne : vendredi 31 mars 2006

Description :

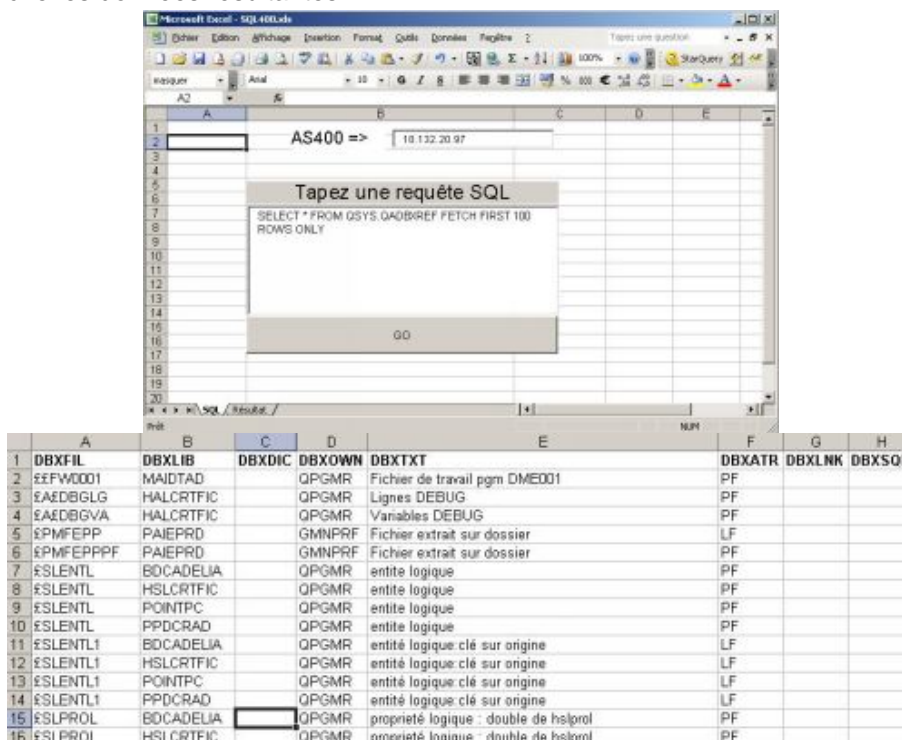
A partir d'une requête SQL saisie dans une feuille Excel, importer des données DB2400.

Environnement iSeries

Voilà un moyen rapide et efficace de s'approprier des données AS400 à partir d'un classeur excel.

Description de l'utilitaire : Cet utilitaire est composé d'un classeur XLS (que vous pouvez télécharger) qui contient 2 feuilles :

- une feuille qui permet de saisir une requête SQL.
- une feuille qui affiche les données résultantes.

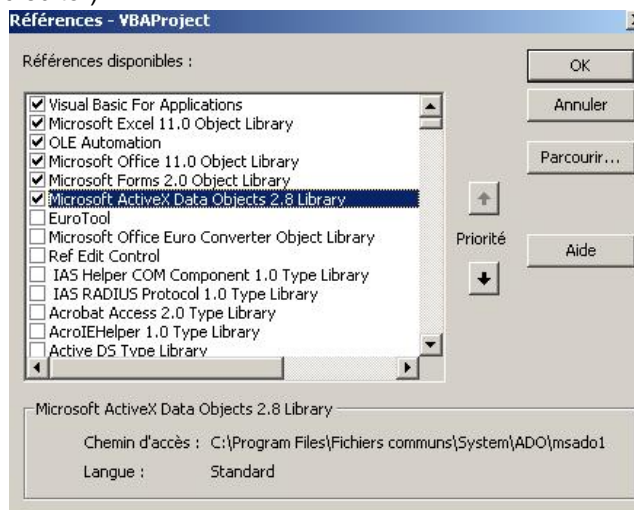


Pré-requis

- Client Access installé sur le poste client
- Microsoft Excel installé sur le poste client

— niveau sécurité macro moyen ou faible (menu outil-macro-sécurité)

— Ajouter référence "Microsoft ActiveX Data Object Library" au classeur (cela ce fait à partir du menu outil-référence dans visual basic editor)



reference

Code source VBA

➤ Procédure liée au bouton de commande :

```
Private Sub CmdGO_Click()  
    Dim wrqt As String  
    Dim was400 As String  
  
    wrqt = UCase(Trim(TextSql.text))  
    was400 = UCase(Trim(TextAS.text))  
    If Len(wrqt) < 10 Then  
        MsgBox "Requête incorrecte !"  
        Exit Sub  
    End If  
    If Left(wrqt, 6) <> "SELECT" Then  
        MsgBox "la requête doit commencer par 'SELECT' !"  
        Exit Sub  
    End If  
    If Len(was400) < 3 Then  
        MsgBox "AS400 incorrect"  
        Exit Sub  
    End If  
  
    ThisWorkbook.Conect wrqt, was400  
End Sub
```

➤ Routine principale :

```
Public Sub Conect(wrqt As String, was400 As String)  
    'on transmet la requête saisie et l'AS400 à connecter  
    Dim Con As New ADODB.Connection  
    Dim Cmd As New ADODB.Command  
    Dim Rs As ADODB.Recordset  
    Dim txtc As String  
    Dim rowCount As Integer  
    Dim colCount As Integer  
    Dim text As String  
    Dim Number As Long  
    Dim val As Variant  
  
    On Error GoTo FINI 'monitorage des erreurs  
    Application.ScreenUpdating = False 'permet d'améliorer les performances  
    Application.Calculation = xlCalculationManual  
  
    '** Indiquer le data source, le user et le Mdp qui vous conviennent  
    '** Si on laisse l'utilisateur et le mot de passe à blanc  
    '** l'identification sera demandée à la lère connection  
    '** txtc = "provider=IBMDA400;data source=" & was400 & "; ;; force translate=297" si probleme de codage  
    caractères  
    txtc = "provider=IBMDA400;data source=" & was400 & "; ;;"  
    Con.Open txtc  
    Set Cmd.ActiveConnection = Con
```

```
Cmd.CommandText = wrqt

Set Rs = Nothing
Set Rs = Cmd.Execute() 'création du curseur
Sheets.Item(2).Rows.Delete 'remise à blanc
rowCount = 1

' Ecriture de l'entête
For colCount = 0 To Rs.Fields.Count - 1
Sheets.Item(2).Cells(rowCount, colCount + 1).Value = Rs.Fields(colCount).Name
Sheets.Item(2).Cells(rowCount, colCount + 1).Font.Bold = True
Next colCount

' Parcours du curseur SQL
While Not Rs.EOF
rowCount = rowCount + 1
For colCount = 0 To Rs.Fields.Count - 1
If Rs.Fields(colCount).ActualSize = -1 Then
text = ""
Else
val = Rs.Fields(colCount).Value
If VarType(val) = vbNull Then
text = ""
Else
text = val
End If
End If
Sheets.Item(2).Cells(rowCount, colCount + 1).Value = text
Next colCount
Rs.MoveNext
Wend

Set Rs = Nothing
Con.Close
Sheets.Item(2).Cells.Columns.AutoFit 'Ajuste les colonnes
Sheets.Item(2).Activate 'donne le focus
Sheets.Item(2).Cells(1, 1).Activate

FINI:
Application.ScreenUpdating = True
Application.Calculation = xlCalculationAutomatic
If Len(Err.Description) > 0 Then ' En cas d'erreur on affiche un message
MsgBox Err.Description
End If
End Sub
```

Téléchargement



SQL400.xls

Post-scriptum :

Importer à partir d'Excel des données ISERIES

La convention d'appellation *SQL est utilisé pour la syntaxe SQL. Vous utiliserez donc le point comme séparateur entre la bib et le fichier (ex : LIB.FICHIER et non pas LIB/FICHIER).