How to use an ActiveX Data Object (ADO) with VBA on a report in Microsoft Dynamics GP

CONFIDENTIAL ARTICLE

(The information in this article is provided to you in accordance with your Confidentiality Agreement)

Article ID : 954619
Last Review : N/A
Revision : 1.1

INTRODUCTION

Sometimes you cannot use Report Writer to create a table relationship to access the data that you want to add to a report. You can use Microsoft Visual Basic for Applications (VBA) to access field in any Microsoft SQL Server table by using ActiveX Data Object (ADO) to connect.

The best practice is as follows:

- Open the connection when the report starts.
- Access the desired data using the before section events.
- Close the connection when the report ends.

When you follow this practice, the connection is available when the report is open. Therefore, the connection is not continuously opened and closed. However, the connection is continuously opened and closed if the connection code is contained in a single event script.

MORE INFORMATION

Assuming that the report fields (or fields in tables already linked to the report) can be used to uniquely identify the desired row in the SQL table, you can create an SQL SELECT statement with a where clause based on the values from the report. You can then create a blank calculated field to return the data from VBA to the report.

Placeholder names have been used in the example code. They are enclosed in braces {}.

The methods of opening the connection to SQL Server are different depending on the version of Microsoft Dynamics GP being used. Example scripts follow:

Method 1: Microsoft Dynamics GP 10.0

Use the following example code:

```
Dim cn As New ADODB.Connection
Dim rst As New ADODB.Recordset
Dim cmd As New ADODB.Command
Dim sqlstring As String

Private Sub Report_Start()
    ' ADO Connection
    Set cn = UserInfoGet.CreateADOConnection
    'Use a client-side cursor so that a recordset count can be obtained later.
    cn.CursorLocation = 3
    'set the database to the currently logged in db
    cn.DefaultDatabase = UserInfoGet.IntercompanyID
End Sub

Private Sub Report_BeforeBody(SuppressBand As Boolean)
    sqlstring = "SELECT {SQLFIELD} FROM {SQLTABLE} WHERE {KEYFIELD} = '" & RTrim({ReportKeyField}) & "'"

    ' ADO Command
    cmd.ActiveConnection = cn
    ' adCmdText
    cmd.CommandType = 1
    ' Command
    cmd.CommandText = sqlstring

   ' Pass through SQL
    Set rst = cmd.Execute
    If Not (rst.EOF And rst.BOF) Then
        {ReportCalcField}.Value = RTrim(rst!{SQLFIELD})
    End If
    rst.Close
End Sub

Private Sub Report_End()
    ' Close ADO Connection
    If rst.State = adStateOpen Then rst.Close
    If cn.State = adStateOpen Then cn.Close
    Set cn = Nothing
    Set rst = Nothing
    Set cmd = Nothing
```

```
End Sub
```

Method 2: Microsoft Dynamics GP 9.0

Use the RetrieveGlobals9.dll file to return an ADO connection object that lets you connect to Microsoft Dynamics GP data.

To download the RetrieveGlobals9.dll file together with its documentation, visit one of the following Microsoft Web sites, depending on whether you are a customer or a partner.

Customer
https://mbs.microsoft.com/customersource/support/documentation/HowToArticles/modifiervbasamples90.htm

Partner
https://mbs.microsoft.com/partnersource/documentation/howtoarticles/modifiervbasamples90.htm

Use the following example code:

```
Dim cn As New ADODB.Connection
Dim rst As New ADODB.Recordset
Dim cmd As New ADODB.Command
Dim sqlstring As String

Private Sub Report_Start()
    Dim userinfo As New RetrieveGlobals9.retrieveuserinfo
    Dim luserid As String
    Dim lintercompanyid As String
    Dim lsqldatasourcename As String
    Dim ldate As Date

    ' RetrieveGlobals
    lsqldatasourcename = userinfo.sql_datasourcename()
    luserid = userinfo.retrieve_user()
    lintercompanyid = userinfo.intercompany_id()
    ldate = CStr(userinfo.user_date())

    ' ADO Connection
    Set cn = userinfo.Connection
    'Use a client-side cursor so that a recordset count can be obtained later.
    cn.CursorLocation = 3
    'set the database to the currently logged in db
    cn.DefaultDatabase = lintercompanyid
End Sub


Private Sub Report_BeforeBody(SuppressBand As Boolean)
    sqlstring = "SELECT {SQLFIELD} FROM {SQLTABLE} WHERE {KEYFIELD} = '" & RTrim({ReportKeyField}) & "'"

    ' ADO Command
    cmd.ActiveConnection = cn
    ' adCmdText
    cmd.CommandType = 1
    ' Command
    cmd.CommandText = sqlstring

   ' Pass through SQL
    Set rst = cmd.Execute
    If Not (rst.EOF And rst.BOF) Then
        {ReportCalcField}.Value = RTrim(rst!{SQLFIELD})
    End If
    rst.Close
End Sub

Private Sub Report_End()
    ' Close ADO Connection
    If rst.State = adStateOpen Then rst.Close
    If cn.State = adStateOpen Then cn.Close
    Set cn = Nothing
    Set rst = Nothing
    Set cmd = Nothing
End Sub
```

Method 3: Microsoft Business Solutions - Great Plains 8.0

Use the RetrieveGlobals.dll file to return an ADO connection object that lets you connect to Microsoft Dynamics GP data.

To download the RetrieveGlobals9.dll file together with its documentation, visit one of the following Microsoft Web sites, depending on whether you are a customer or a partner.

Customer
https://mbs.microsoft.com/customersource/support/documentation/HowToArticles/ModifierVBASamples80.htm

Partner
https://mbs.microsoft.com/partnersource/documentation/howtoarticles/modifierVBASamples80.htm

Use the following example code:

```
Dim cn As New ADODB.Connection
Dim rst As New ADODB.Recordset
Dim cmd As New ADODB.Command
Dim sqlstring As String

Private Sub Report_Start()
    Dim userinfo As New RetrieveGlobals.retrieveuserinfo
    Dim luserid As String
    Dim lintercompanyid As String
    Dim lsqldatasourcename As String
    Dim lsqlpassword As String
    Dim constring As String

    ' RetrieveGlobals
    lsqldatasourcename = userinfo.sql_datasourcename()
    luserid = userinfo.retrieve_user()
    lsqlpassword = userinfo.sql_password()
    lintercompanyid = userinfo.intercompany_id()
    'MsgBox (luserid & " " & lsqlpassword & " " & lintercompanyid & " " & lsqldatasourcename)

    ' Create Connection String
    constring = "Provider=MSDASQL" & _
                ";Data Source=" & lsqldatasourcename & _
                ";User ID=" & luserid & _
                ";Password=" & lsqlpassword & _
                ";Initial Catalog=" & lintercompanyid
    'MsgBox constring

    ' ADO Connection
    With cn
        .ConnectionString = constring
        .CursorLocation = 3 ' adClient
        .Open
    End With
End Sub


Private Sub Report_BeforeBody(SuppressBand As Boolean)
    sqlstring = "SELECT {SQLFIELD} FROM {SQLTABLE} WHERE {KEYFIELD} = '" & RTrim({ReportKeyField}) & "'"

    ' ADO Command
    cmd.ActiveConnection = cn
    ' adCmdText
    cmd.CommandType = 1
    ' Command
    cmd.CommandText = sqlstring

   ' Pass through SQL
    Set rst = cmd.Execute
    If Not (rst.EOF And rst.BOF) Then
        {ReportCalcField}.Value = RTrim(rst!{SQLFIELD})
    End If
    rst.Close
End Sub

Private Sub Report_End()
    ' Close ADO Connection
    If rst.State = adStateOpen Then rst.Close
    If cn.State = adStateOpen Then cn.Close
    Set cn = Nothing
    Set rst = Nothing
    Set cmd = Nothing
End Sub
```

REFERENCES

For more information about using ADO with VBA on a window, click the following article number to view the article in the Microsoft Knowledge Base:

942327 How to use ActiveX Data Object (ADO) with VBA on a window with Microsoft Dynamics GP and with Microsoft Business Solutions - Great Plains 8.0

For more information about printing issues with a graphical report, click the following article number to view the article in the Microsoft Knowledge Base:

884601 The printed version of a graphical report may not include all the data when you print the graphical report directly to a printer in Microsoft Dynamics GP

APPLIES TO

- Modifier with Visual Basic for Applications
- Report Writer, when used with:
  Microsoft Dynamics GP 10.0
  Microsoft Dynamics GP 9.0
  Microsoft Business Solutions–Great Plains 8.0

Keywords: kbmbsvba kbexpertiseadvanced kbmbspartner kbmbsmigrate kbhowto kbexpertiseinter KB954619