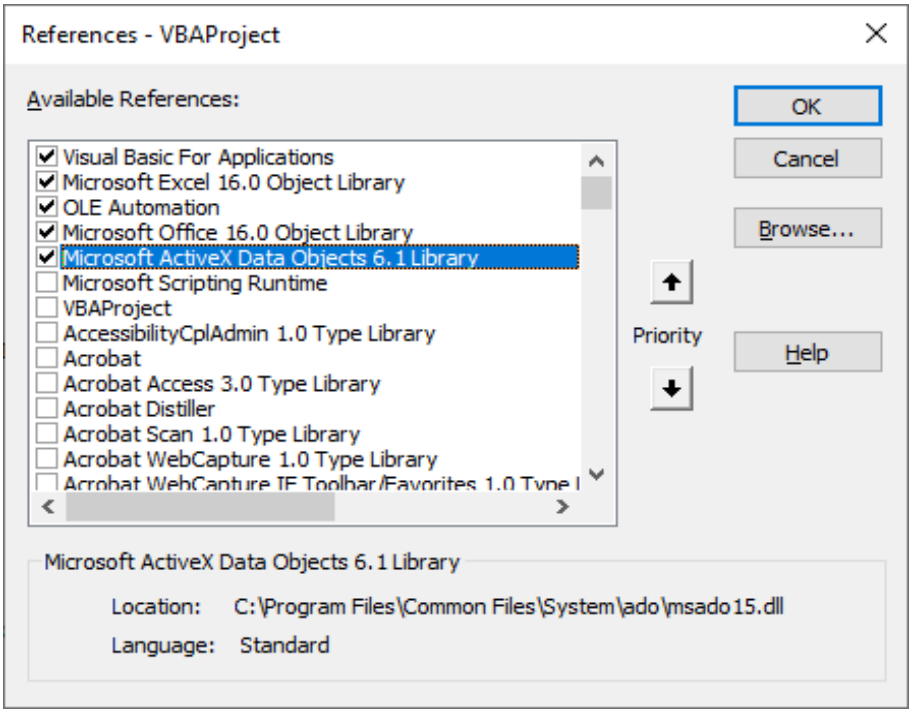
How to Connect to SQL Server in Microsoft Excel

As of 8/21/22

Library Reference

The latest library to use is "**Microsoft ActiveX Data Objects 6.1 Library**". Add that library as a reference.



Connection Strings

The connection string to use is as follows:

```
Public Const cAPPDBSTRING As String =
"Provider=SQLNCLI11;Server=localhost\SQLDEVELOPER19;Database=vivedb;Trusted_Connection=Yes;"
```

You can use "localhost" or the "computer name" since you are pointing to a database on the same computer as your application.

Other connection strings that can be used include:

Connection string to use for computers with the 'SQLNCLI11.DLL' driver:

Public Const cAPPDBSTRING_SQLN1 As String = "Provider=SQLNCLI11;Server=W7-PC07GA3L\SQLEXPRESS;Database=kbalocaldb;Trusted_Connection=Yes;"

Public Const cAPPDBSTRING_SQLN2 As String = "Provider=SQLNCLI11;Server=AWILL-LAPTOP-L\SQLEXPRESS;Database=kbalocaldb;Trusted_Connection=Yes;"

Public Const cAPPDBSTRING_SQLN3 As String = "Provider=SQLNCLI11;Server=localhost;Database=PFIL;Trusted_Connection=Yes;"

Connection string to use for computers with the 'SQLSRV32.DLL' driver:

Public Const cAPPDBSTRING_OLEDB1 As String = "Provider=SQLOLEDB;Server=184.168.47.15;Initial Catalog=kbalocaldb;User ID=awcdevelopment;Password=mypassword;"

These also works somehow:

Public Const cAPPDBSTRING_OTHER1 As String = "Driver={SQL Server};Server=184.168.47.15;Initial Catalog=kbalocaldb;User ID=awcdevelopment;Password=awcwebdbPaSSw0rd!;"

Public Const cAPPDBSTRING_OTHER2 As String = "Provider=MSDASQL;DSN=GreatPlains;Initial Catalog=TWO;User Id=sa;Password=password"

Other Information

To find out which providers you have on your system, you can run this stored procedure in SQL Server.

EXEC master.dbo.sp_MSset_oledb_prop

ADsDSOObject

Microsoft.ACE.OLEDB.12.0

Microsoft.ACE.OLEDB.16.0

MSDAOSP

MSDASQL:MSDASQL is Microsoft's Object Linking and Embedding Database (OLE DB) provider for Open Database Connectivity (ODBC) drivers. MSDASQL allows OLE DB consumer applications to use ODBC drivers to connect to a data source. The provider converts incoming OLE DB calls into ODBC calls and passes them on to the specified ODBC driver. The provider then retrieves results from the ODBC driver and formats the results into OLE DB specific structures, which the user can then access.

MSOLAP

MSOLEDBSQL

Search.CollatorDSO

SQLNCLI11

SQLOLEDB: SQLOLEDB is the Microsoft OLE DB Provider for SQL Server that allows ADO to access Microsoft SQL Server. The provider remains deprecated and it is not recommended to use it for new development work. Instead, use the new Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL) which will be updated with the most recent server features.

SSISOLEDB

Difference between Driver, Provider, Library Reference, and Connection String

The Provider (e.g. SQLNCI11, MSOLEDBSQL, MSDASQL) is a set of libraries (DLL files) that handles the connection and communication back and forth between your application and the database.

However, once you're connected and are communicating with the database, how do you then interact with the database by manipulating the objects and data in the database. For example, how to you read, add, update, and delete records? How do you add, drop tables? Etc… That is where the Driver comes in. The Driver (e.g. ADO, OLEDB, ODBC) is a library (DLL file) that allows you to interact with a database to manipulate the database objects and records.

In sum,

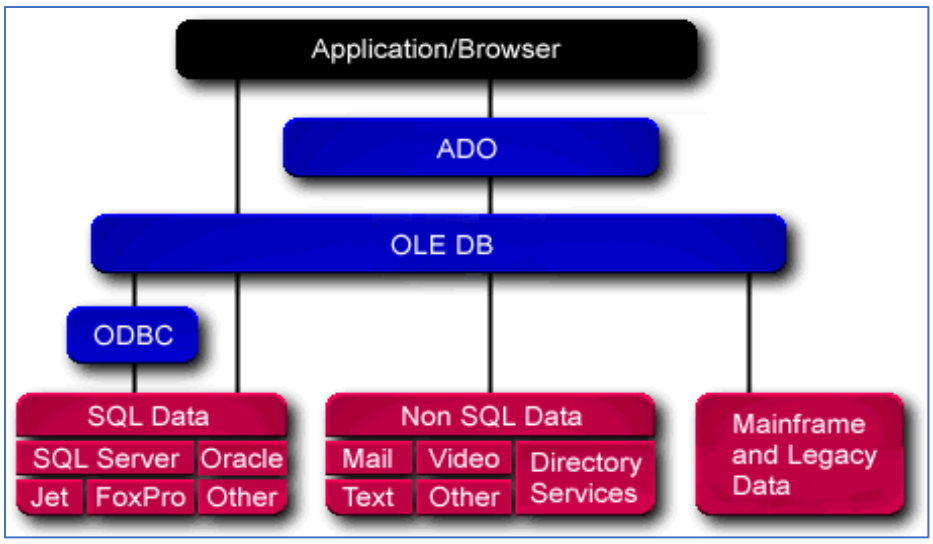
- The Provider handles the connection and communication of commands back and forth between your application.
- The Driver handles the interaction with the database to manipulate the database objects and records.

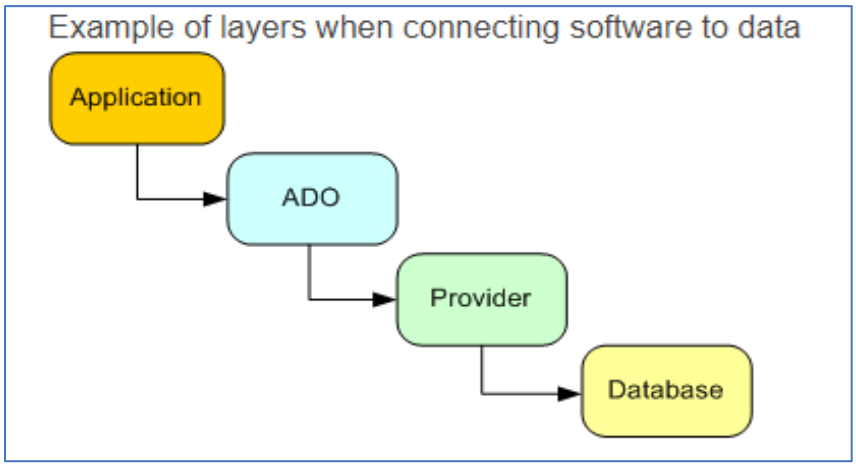
What about the Library Reference and Connection String?

The **Library Reference** is what will allow you to use the **Driver** in your application. In other words, to be able to use a **Driver**, you must first **reference** it in your application so that you can use the objects and associated methods and objects of that **Driver** in your application. It's just as it states… a "**reference**", so your application can find the **Driver** and the things within that **Driver**. Note that this has nothing to do with your **Provider**. This is strictly related to the **Driver**.

The **Connection String** is what you use to send information to the **Provider** so that you can connect and interact with a database. It contains the name of the **Provider** you want to use along with information about the database you want to connect to (e.g. database name, server name, IP address, user name, password, etc…). Here, the **Driver** has nothing to do with it, it's strictly related to the Provider.

Diagrams





References

https://www.connectionstrings.com

https://stackoverflow.com/questions/19293744/difference-between-driver-and-provider

Script Using ADO

```
Public Const cAPPDBSTRING As String =
"Provider=SQLNCLI11;Server=localhost\SQLDEVELOPER19;Database=testdb;Trusted_Connection=Yes;"
```

```
Sub SQLServer_CreateRecord()
  'Declare and set local variables.
    Dim vCN As ADODB.Connection, vCM As ADODB.Command, vRS As ADODB.Recordset
    Dim vCS As String, vSQ As String

    Set vCN = New ADODB.Connection
    Set vCM = New ADODB.Command
    Set vRS = New ADODB.Recordset

  'Open database connection.
    vCS = cAPPDBSTRING
    vCN.Open vCS

  'Set command object.
    vCM.ActiveConnection = vCN
    vCM.CommandType = adCmdText
    vCM.CommandText = "INSERT INTO [MYTABLE] (MYTABLE_NAME) VALUES ('My Name')"

  'Execute command.
    vCM.Execute

  'Close database connection and clear objects.
    vCN.Close
    Set vRS = Nothing
    Set vCM = Nothing
    Set vCN = Nothing

  'All done.
    MsgBox "All done!"
End Sub
```

```
Sub SQLServer_ReadRecord()
  'Declare and set local variables.
    Dim vCN As ADODB.Connection, vCM As ADODB.Command, vRS As ADODB.Recordset
    Dim vCS As String, vSQ As String

    Set vCN = New ADODB.Connection
    Set vCM = New ADODB.Command
    Set vRS = New ADODB.Recordset

  'Open database connection.
    vCS = cAPPDBSTRING
    vCN.Open vCS

  'Set command object.
```

```
    vCM.ActiveConnection = vCN
    vCM.CommandType = adCmdText
    vCM.CommandText = "SELECT * FROM [MYTABLE]"

  'Execute command.
    Set vRS = vCM.Execute()

    If Not (vRS.BOF And vRS.EOF) Then
       Debug.Print vRS.Fields(0), vRS.Fields(1)
    End If

  'Close database connection and clear objects.
    vCN.Close
    Set vRS = Nothing
    Set vCM = Nothing
    Set vCN = Nothing

  'All done.
    MsgBox "All done!"
End Sub
```

```
Sub SQLServer_UpdateRecord()
  'Declare and set local variables.
    Dim vCN As ADODB.Connection, vCM As ADODB.Command, vRS As ADODB.Recordset
    Dim vCS As String, vSQ As String

    Set vCN = New ADODB.Connection
    Set vCM = New ADODB.Command
    Set vRS = New ADODB.Recordset

  'Open database connection.
    vCS = cAPPDBSTRING
    vCN.Open vCS

  'Set command object.
    vCM.ActiveConnection = vCN
    vCM.CommandType = adCmdText
    vCM.CommandText = "UPDATE [MYTABLE] SET MYTABLE_NAME = 'My Name Updated' WHERE MYTABLE_ID = 1"

  'Execute command.
    vCM.Execute

  'Close database connection and clear objects.
    vCN.Close
    Set vRS = Nothing
    Set vCM = Nothing
    Set vCN = Nothing

  'All done.
```

```
   MsgBox "All done!"
End Sub
```

```
Sub SQLServer_DeleteRecord()
  'Declare and set local variables.
    Dim vCN As ADODB.Connection, vCM As ADODB.Command, vRS As ADODB.Recordset
    Dim vCS As String, vSQ As String

    Set vCN = New ADODB.Connection
    Set vCM = New ADODB.Command
    Set vRS = New ADODB.Recordset

  'Open database connection.
    vCS = cAPPDBSTRING
    vCN.Open vCS

  'Set command object.
    vCM.ActiveConnection = vCN
    vCM.CommandType = adCmdText
    vCM.CommandText = "DELETE [MYTABLE] WHERE MYTABLE_ID = 1"

  'Execute command.
    vCM.Execute

  'Close database connection and clear objects.
    vCN.Close
    Set vRS = Nothing
    Set vCM = Nothing
    Set vCN = Nothing

  'All done.
    MsgBox "All done!"
End Sub
```

```
Sub SQLServer_StoredProcedure()
  'Declare and set local variables.
    Dim vCN As ADODB.Connection, vCM As ADODB.Command
    Dim vCS As String, vSQ As String

    Set vCN = New ADODB.Connection
    Set vCM = New ADODB.Command

  'Open database connection.
    vCS = cAPPDBSTRING
```

```
    vCN.Open vCS

  'Set command object.
    vCM.ActiveConnection = vCN

    vCM.Parameters.Append vCM.CreateParameter(Name:="pUserID", Type:=adInteger, Direction:=adParamInput,
Value:=1)
    vCM.Parameters.Append vCM.CreateParameter(Name:="pPlanTemplateIndex", Type:=adVarChar,
Direction:=adParamInput, Size:=300, Value:=1)
    vCM.Parameters.Append vCM.CreateParameter(Name:="pPlanTemplateIsPublished", Type:=adBoolean,
Direction:=adParamInput, Value:=1)
    vCM.Parameters.Append vCM.CreateParameter(Name:="pRecordLog", Type:=adBoolean, Direction:=adParamInput,
Value:=1)
    vCM.Parameters.Append vCM.CreateParameter(Name:="pOutputMessage", Direction:=adParamOutput,
Type:=adVarChar, Size:=300)
    vCM.CommandType = adCmdStoredProc
    vCM.CommandText = "spPlanTemplatePublish"

  'Execute command.
    vCM.Execute

  'Close database connection and clear objects.
    vCN.Close
    Set vCM = Nothing
    Set vCN = Nothing
End Sub
```