Instructor Notes:

Add instructor notes here.

Instructor Notes:

Add instructor notes here.

Lesson Objectives

- ADO Collections
  - Fields
  - Properties
  - Parameters
  - Errors
- ADO Objects
  - Connection
  - Command
  - Parameter
  - Recordset
  - Field
  - Error
  - Property

Instructor Notes:

Add instructor notes here.

Lesson Objectives

- Recordset Object.
  - Querying a database.
  - Filtering records.
  - Searching for records.
  - Sorting records.
  - Navigating in a recordset.
  - Adding new records.
- File System Object.
  - File Object Properties
  - File Object Methods

Instructor Notes:

ADODB Object Model

- ADODB is a COM object that enables applications to gain access to, and modify a wide variety of data sources.
- The typical data source is a relational database that supports the Open Database Connectivity (ODBC) standard and is manipulated with commands written in Structured Query Language (SQL).

Instructor Notes:

Basic ADO Programming Model

ADO provides the means for you to perform the following sequence of actions:

- Create a connection object to connect to the datasource.
- Create a recordset object in order to receive data in.
- Open the connection
- Populate the recordset by opening it and passing the desired table name or SQL statement as a parameter to *open* function.
- Do all the desired searching/processing on the fetched data.
- Commit the changes you made to the data (if any) by using *Update* or *UpdateBatch* methods.
- Close the recordset
- Close the connection

Instructor Notes:

ADO Collections

Collections :

- **Errors** - All the **Error** objects created in response to a single failure on a connection.
- **Parameters** - All the Parameter objects associated with a Command object.
- **Fields** - All the **Field** objects associated with a **Recordset** object.
- **Properties** - All the **Property** objects associated with a **Connection**, **Command**, **Recordset** or **Field** object.

Instructor Notes:

ADO Objects

Objects :

- **Connection** - Enables exchange of data.
- **Command** - Embodies an SQL statement.
- **Parameter** - Embodies a parameter of an SQL statement
- **Recordset** - Enables navigation and manipulation of data.
- **Record** - This object represents one record in the database and contains a fields collection.
- **Field** - Embodies a column of a **Recordset** object.
- **Error** - Embodies an error on a connection.
- **Property** - Embodies a characteristic of an ADO object.

Connection - The connection object stores information about the session and provides methods of connecting to the data store. A connection object connects to the data store using its 'Open' method with a connection string which specifies the connection as a list of key value pairs.
Command - After the connection object establishes a session to the data source, instructions are sent to the data provider via the command object. The command object can send SQL queries directly to the provider through the use of the CommandText property, send a parameterised query or stored procedure through the use of a Parameter object or Parameters collection or run a query and return the results to a dataset object via the Execute method.
Recordset - A recordset is a group of records. It contains a Fields collection and a Properties collection. The Fields collection is a set of Field objects, which are the corresponding columns in the table. The Properties collection is a set of Property
objects, which defines a particular functionality of an OLE DB provider.
Parameter - A parameter is a means of altering the behavior. For instance a stored procedure might have different parameters passed to it depending on what needs to be done. These are called parameterised commands.
Field - Each Record object contains many fields. Each field corresponds to a column in the database table that it references.
Property -This object is specific to the OLE DB provider and defines an ability that the provider has implemented. A property object can be either a built-in property — it is a well-defined property implemented by ADO already and thus cannot be altered — or can be a dynamic property — defined by the underlying data provider and can be changed
Error - When an OLE DB provider error occurs during the use of ADO, an Error object will be created in the Errors collection.

Instructor Notes:

Manipulating the Recordset Object

You use Recordset objects to manipulate data from a provider. This allows you to :

- Query a database
- Filtering a recordset
- Searching a recordset
- Sorting the records
- Navigating a recordset
- Adding a new record : Usually, in testing, we don’t add new records to the database.

Instructor Notes:

Querying a Database

Syntax:

```
recordset.Open Source, ActiveConnection, CursorType, LockType, Options
```

- *CursorType* - Indicates the type of cursor used in a Recordset object.
  - **adOpenForwardOnly** – Default. You can only scroll forward through records.
  - **adOpenKeyset** - you can't see records that other users add, although records that other users delete are inaccessible from your recordset.
  - **adOpenDynamic** - Additions, changes, and deletions by other users are visible, and all types of movement through the recordset are allowed.
  - **adOpenStatic** - A static copy of a set of records that you can use to find data or generate reports. Additions, changes, or deletions by other users are not visible.

Instructor Notes:

Querying a Database

- *LockType - Indicates the type of locks placed on records during editing.*
  - **adLockReadOnly** – Default. Read-only. In testing, when we reading data, we want to be sure that we are not locking the application. The *recommended locktype for testing, is using the flag* ***adLockReadOnly.***
  - **adLockOptimistic** - the provider uses optimistic locking, locking records only when you call the Update method.
- *Options - indicates how the provider should evaluate the Source argument.*
  - **adCmdText** - textual definition of an SQL command
  - **adCmdTable** - Indicates that ADO should generate an SQL query to return all rows from the table named in ***Source***.
  - **adCmdTableDirect** - Indicates that the provider should return all rows from the table named in ***Source***.
  - **adCmdStoredProc -** Indicates that the provider should evaluate ***Source*** as a stored procedure.

Instructor Notes:

Querying a database – Example

```
strSQL = "Select * From Employes Where State='NY'"
Set objRst = objConn.Execute(strSQL)
'instead of the above one line we can also write the below two lines
'Set objRst = CreateObject("ADODB.Recordset")
'objRst.Open strSQL, objConn
While Not objRst.EOF
        iCounter = iCounter + 1
        strRep = "First Name : " & objRst("EmpFirstName").Value & vbCr
        strRep = strRep  & "Last Name : " & objRst("EmpLastName").Value & vbCr
        strRep = strRep  & "Address: " & objRst("EmpAddress").Value & vbCr
        MsgBox strRep,0,"Employee " & iCounter
        objRst.MoveNext
Loop
```

Instructor Notes:

Demo

- dbConnectivity

Instructor Notes:

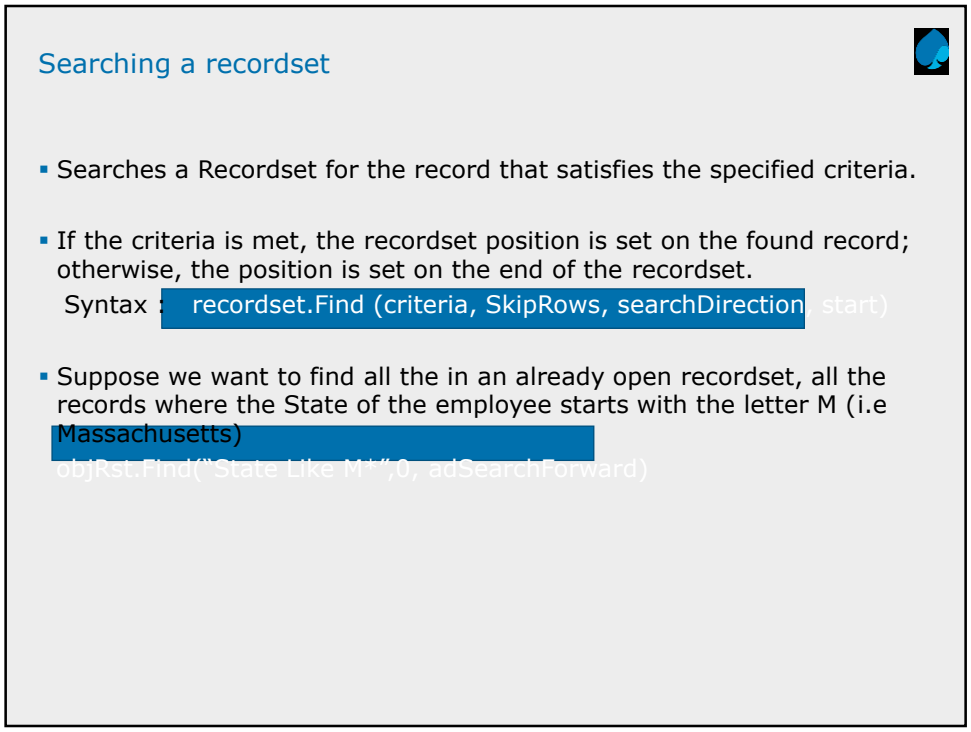
Filtering a recordset

- For this sample, we have already open the record and selected all the employees.
- Now we can execute the follow:

```
objRst.Filter = "City = 'New York' Or City = 'Washington'"
```

- The new recordset will contain the filtered records.

Instructor Notes:



Criteria - A **String** containing a statement that specifies the column name, comparison operator, and value to use in the search.
SkipRows - An optional **Long** value, whose default value is zero, that specifies the offset from the current row or ***start*** bookmark to begin the search.
searchDirection - An optional **SearchDirectionEnum** value that specifies whether the search should begin on the current row or the next available row in the direction of the search. Its value can be **adSearchForward** or **adSearchBackward**. The search stops at the start or end of the recordset, depending on the value of ***searchDirection***.
start - An optional **Variant** bookmark to use as the starting position for the search.

Instructor Notes:

Navigating in a recordset

- We can navigate in a recordset in any desired direction using the MoveFirst, MoveLast, MoveNext, and MovePrevious Methods.
- You can't navigate to previous or first if you open the recordset adOpenForwardOnly flag.

Example : objRst.MoveFirst, objRst.MoveNext

Instructor Notes:

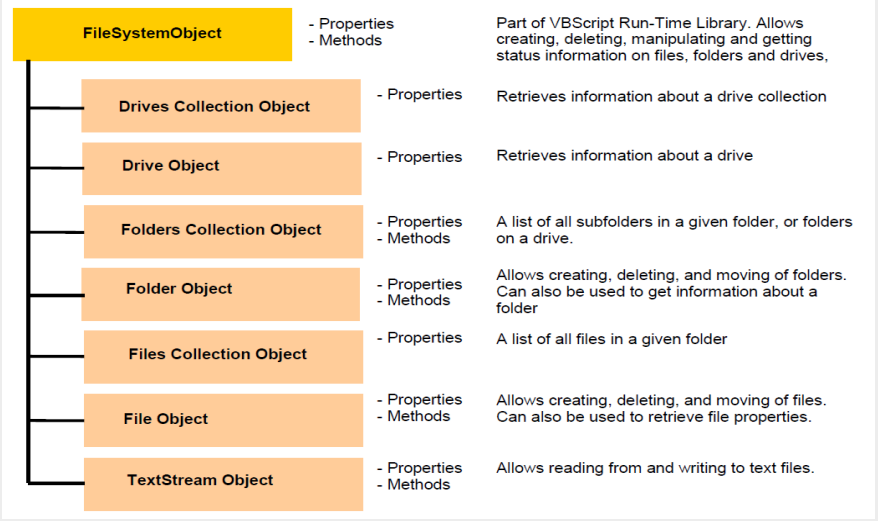
File System Object

- Many a times during testing you may need to interact with Drives, folder and text files using QTP.
- Interaction can be(but not limited to) in the form of :
  - Creating a file : in testing, we usually do not create any file
  - reading input from a file : In testing we test the data and reading from the file is what we usually do during testing.
  - writing output to a file.
  - Deleting a file
- The FileSystemObject (FSO) provides an API to access the Windows filesystem, providing access to files, drives, text streams

Instructor Notes:

FSO Model

FSO object model contains the following objects and collections.



Instructor Notes:

File Object - Properties

Property	Description
Attributes	Returns number of attributes supported by a particular file.
DateCreated	Returns the date along with the time of the creation of a particular file
DateLastAccessed	Returns date along with the time when a particular file was last accessed
DateLastModified	Returns the date along with the time when a particular file was last modified
Drive	Returns the drive in which the particular file is located.
Name	Returns the name of a particular file.
ParentFolder	Returns the parent folder of a particular file as if the file is stored in C drive so it will return C:\
Path	Returns the path of a particular file as if the file is stored in C drive and the name of the file is a test so it will return C:\myfile.txt.
Size	Returns the size of a particular file in bytes.
Type	Returns the type of a particular file i.e. file type description like a file which ends with .vbs, for that “VBScript” will be returned.

Instructor Notes:

Demo

- fileProperties

Instructor Notes:

File Object - Methods

Method	Description
obj.CopyFile srcpath, destpath	copies the mentioned file/folder to a specific destination.
obj.DeleteFile fileNameAlongWithPath	Deletes a particular file
obj.MoveFile srcpath, destpath	Move a particular file to new destination
OpenTextFile(filename, iomode, format)	Open a file in different modes. If you want to open a text file for reading only then you can pass the constant value for iomode as **1**, **2** in case of writing and **8** for appending purpose. The optional format can be set to Tristate Constants - -1 for opening file in Unicode, 0 to open as ASCII and -2 to open file in default system setting.
CreateTextFile(fileName, overwrite, Unicode)	Creates a text file. The optional overwrite parameter returns a Boolean value - True (the default) permits overwriting of existing files while False does not. The optional parameter, unicode, is a Boolean. In this case, True creates a Unicode file and False (the default) creates an AscII file
FileExists(file_path)	Checks if the mentioned file exists or not. Returns TRUE if exists else FALSE.
GetFile FetFileName	Get the mentioned file.

Instructor Notes:

Reading Contents from a File

Syntax :

Object.ReadLine()

```
Public Sub ReadFile()
  Dim fso, MyFile, data
  Const ForReading = 1

  Set fso = CreateObject("Scripting.FileSystemObject")
  Set myfile = fso.OpenTextFile("C:\testresults.txt", ForReading, True)

  Do while myfile.AtEndOfStream <> True
      data = Myfile.ReadLine()
      Msgbox data
  Loop
End Sub
```

Instructor Notes:

Demo

- openReadFromFile
- createFile
- writeToFile
- writeToExcel
- Student.vbs

Instructor Notes:

Lab

Lab 7

Instructor Notes:

Lesson Summary

- After completing this module, you now know about
- Database connectivity using VBScript
  - Connecting to database, opening recordset, Querying, navigating
- File System Object

Instructor Notes:

1. Option3:\?
2. Event Handler
3. Links

Review - Questions

- Question 1: What is the need of ADO Collections ?
- Question 2: List the ADO objects.
- Question 3: State the difference between Recordset and record.
- Question 4: State the syntax of querying a database and explain its parameters.