

Class: MSc

Subject : Application of IT- Basics and Advance Excel

Chapter: Unit 4 Chapter 5

Chapter Name: Opening Files

Handling other files in VBA macros

- *As a programmer, you're likely to come across many situations where other files need to be opened for reading, writing, or appending data. A macro can help you easily handle these files, and the Open method is your ticket.*
- *VBA offers simple method to open and work on files. This permits a user to either read or write — or do both — after opening the file.*

Handling other files in VBA macros

Syntax: `Open <path name> For <mode> As <FiTenumber>`

Explaining the above parameters:

<path name> : A mandatory field. It is the name of the file along with details of its extension, drive and directory.

<mode> : This is a mandatory field. It can be any of the five options below:

Append: The application writes data to the end of the file.

Binary: The data in the file can be read and written in bytes. Using bytes means that the data can be any type.

Output: The application writes data to the file. Any existing data in the file are deleted when the file is opened.

Input: The application reads data from a file.

Random – This is the default mode.

Filenumber: The FiTenumber is also called file handle and can range from 1 to 511 inclusive. This enables hundreds of different files to be open at the same time, all with unique identification numbers. File handles can be specified explicitly, like this:

```
Open MyDataFile For Input As 5
```

Note: Opening a file in Output mode deletes all the data from the file, whereas opening a file in Append mode preserves any data already in the file.

Example 1: Just open a file

- This is a simple program which simply opens an Excel file.

```
Sub open_file_demo()  
  
    ' declare variable  
    Dim pathname  
  
    ' assign a value  
    pathname = "C:\Users\jaine\OneDrive\Documents\Test file.xlsx"  
    ' now open the file using the open statement  
    Workbooks.Open pathname  
  
End Sub
```

Example 2

Open an Excel workbook in "read only" mode and try to write to it

```
Sub open_file_demo2()  
  
    ' declare variables  
    Dim pathname  
    Dim wb As Workbook  
  
    ' assign a value  
    pathname = "C:\Users\jaine\OneDrive\Documents\Test file.xlsx"  
  
    ' now open the file using the open statement and assign it to the wb object so that it can be  
    Set wb = Workbooks.Open(Filename:=pathname, ReadOnly:=True)  
  
    ' Try writing after opening the file in "read only mode". This should throw an error.  
    wb.Sheets(0).Cells(1, 1).Value = "Try writing"  
  
End Sub
```

Example 3

Open a text file and read its contents using the Open function and file number

```
Sub TextFile_PullData()  
  
    ' declare variables  
  
    Dim int_txtfile As Integer  
    Dim str_file_Path As String  
    Dim str_File_Content As String  
  
    ' Assign the file path to the variable  
    str_file_Path = "C:\Users\jaine\OneDrive\Documents\Sample.txt"  
  
    ' Find the next available file number to be used by the fileopen fun  
    int_txtfile = FreeFile  
  
    ' Open the said text file using the function  
    Open str_file_Path For Input As int_txtfile  
  
    ' The content of the file is stored in a variable  
    str_File_Content = Input(LOF(int_txtfile), int_txtfile)  
  
    ' Print the file content using the variable in which it is stored  
    Debug.Print str_File_Content  
  
    ' Close the opened text file  
    Close int_txtfile  
  
End Sub
```

Example 5

In this example, we will add additional text to the end of an existing text file that already has some data.

```
Sub txt_file_append()  
  
    ' declare variables  
  
    Dim int_txtfile As Integer  
    Dim str_file_Path As String  
    Dim str_File_Content As String  
  
    ' Assign the file path to the variable  
    str_file_Path = "C:\Users\jaine\OneDrive\Documents\Sample.txt"  
  
    ' Find the next available file number to be used by the fileopen function  
    int_txtfile = FreeFile  
  
    ' Open the said text file using the function  
    Open str_file_Path For Append As int_txtfile  
  
    ' Write content to the end of the file  
    Print #int_txtfile, "This is additional content"  
    Print #int_txtfile, "Warm Regards"  
  
    ' Close the opened text file  
    Close int_txtfile  
  
End Sub
```