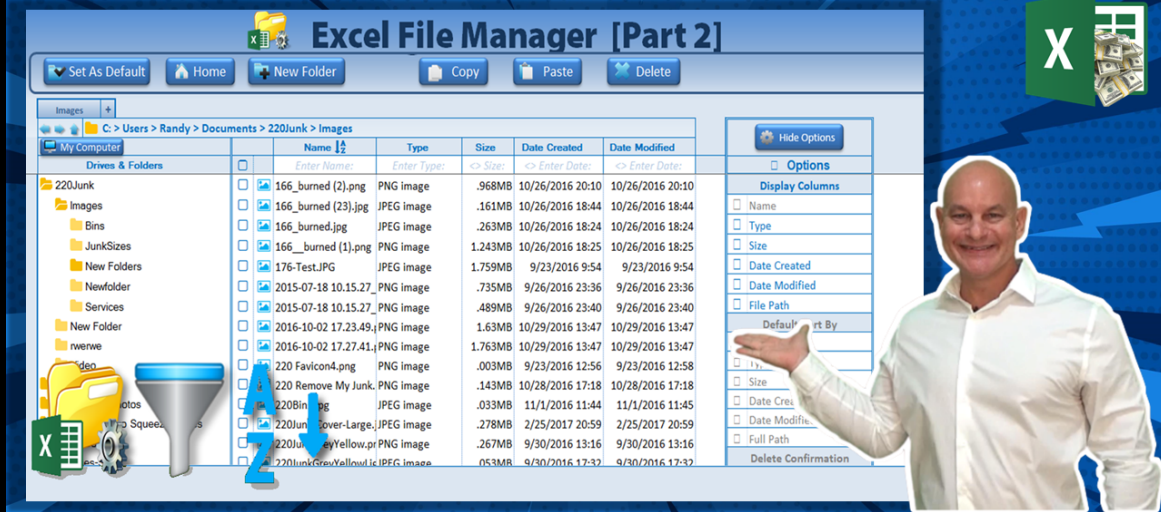


VBA SOURCE CODE BOOK

FILE MANAGER PT. 2



Excel For Freelancers

How To Create this Excel File Manager, with Filtering, Sorting and Adding Folders [Part 2]



[DOWNLOAD
APPLICATION](#)



[WATCH ENTIRE
SERIES](#)



[VIEW
TRAINING](#)

by: Randy Austin

ABOUT THE AUTHOR

A two-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With nearly 300,000 YouTube subscribers, 14,000,000 video views, 200+ comprehensive training videos, and a thriving 40,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



Microsoft®
Most Valuable
Professional



OUR COURSES & PRODUCTS



This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



Incredible Package of 175 of my BEST Applications into a SINGLE ZIP File which also includes the "175 Workbook Library".

[Click here to learn more](#)



With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)

Projects	2
VBAProject	2
Documents	2
Sheet1	2
(Declarations).....	2
Worksheet_Change [Sub]	2
Worksheet_SelectionChange [Sub]	2
Sheet2	5
(Declarations).....	5
Sheet3	7
(Declarations).....	7
ThisWorkbook	9
(Declarations).....	9
Modules	11
CodeRuntimeMacros	11
(Declarations).....	11
ResetCalc [Sub]	11
StartClock [Sub]	11
StopCalc [Sub]	11
StopClock [Sub]	11
DrivesMacros	13
(Declarations).....	13
DisplayDrives [Sub]	13
FadeOutMessages	15
(Declarations).....	15
FoldDefaultMess [Sub]	15
FileMacros	17
(Declarations).....	17
CopySelectedFiles [Sub]	17
DeleteSelectedFiles [Sub]	17
PasteSelectedFiles [Sub]	17
ReloadFolder [Sub]	18
SelectedFilesFilter [Sub]	18
FilterMacros	20
(Declarations).....	20
Filter_Clear [Sub]	20
Filter_Run [Sub]	20
FolderMacros	22
(Declarations).....	22
CloseFolder [Sub]	22
CreateNewFolder [Sub]	23
ExpandFolder [Sub]	23
GoToHomeFolder [Sub]	25
GoToNextFolder [Sub]	25
GoToParentFolder [Sub]	25
GoToPrevFolder [Sub]	25
OpenFolder [Sub]	25
SetAsDefaultFolder [Sub]	26
OptionMacros	28
(Declarations).....	28
HideOptions [Sub]	28
ShowHideColumns [Sub]	28
ShowOptions [Sub]	28
SortingMacros	30
(Declarations).....	30
FileSort [Sub]	30

```

1 Option Explicit
2
3 Dim objFSO, objFolder As Object
4
5 Private Sub Worksheet_Change(ByVal Target As Range)
6     If Target.Cells.CountLarge > 1 Then Exit Sub
7     'Change Folder Name
8     If Not Intersect(Target, Range("D9:O1000" )) Is Nothing And Range("B2" ).Value = False
9         Then 'On Folder Name Change
10            If Target.Offset(0, -1).Value = Sheet2.Range("OpenFold" ).Value Or Target.Offset(0, -1)
11                .Value = Sheet2.Range("CloseFold" ).Value Then
12                Dim OldFolderName, NewFolderName As String
13                Set objFSO = CreateObject("Scripting.FileSystemObject" )
14                Set objFolder = objFSO.GetFolder(Range("B" & Target.Row).Value)
15                OldFolderName = Range("B" & Target.Row).Value 'Old Folder Name
16                NewFolderName = objFolder.ParentFolder & "\" & Target.Value
17                'On Error GoTo FileOpenMsg
18                Name OldFolderName As NewFolderName
19                Range("B" & Target.Row).Value = NewFolderName 'Update Path
20                Application.SendKeys "{numlock}%s" , True 'Reset Number Lock on Keyboard
21            End If
22        End If
23
24        'Run Filter on cell change
25        If Not Intersect(Target, Range("S7:W7" )) Is Nothing Then
26            On Error Resume Next 'Bug's can appear on Clearing filter if this is not added
27            If InStr(Target.Value, ":" ) = 0 Then Filter_Run
28        End If
29
30        If Not Intersect(Target, Range("S9:S1000" )) Is Nothing And Range("B2" ).Value = False
31            Then 'On File name change but not on load files
32                If Target.Value <> Empty Then
33                    Set objFSO = CreateObject("Scripting.FileSystemObject" )
34                    objFSO.MoveFile Range("X" & Target.Row).Value, Range("B5" ).Value & "\" & Target.
35                        Value
36                    Range("X" & Target.Row).Value = Range("B5" ).Value & "\" & Target.Value 'Update
37                        File Path
38                End If
39            End If
40        End If
41        Exit Sub
42        FileOpenMsg:
43        MsgBox "Folder cannot be renamed becuse a file or folder may be open in another program
44            or file manager"
45    End Sub
46
47 Private Sub Worksheet_SelectionChange(ByVal Target As Range)
48     If Target.Cells.CountLarge > 1 Then Exit Sub
49     'On Selection Change of Folder area
50     If Not Intersect(Target, Range("D9:O1000" )) Is Nothing And Range("B2" ).Value = False
51         Then 'On Folder Selection
52             If Target.Value = Sheet2.Range("CloseFold" ).Value Then ExpandFolder
53             If Target.Value = Sheet2.Range("OpenFold" ).Value Then CloseFolder
54             If Target.Value = Sheet2.Range("DriveIcon" ).Value Then
55                 Range("B5" ).Value = Range("B" & Target.Row).Value
56                 OpenFolder
57             End If
58         End If
59     End If
60     'On Selection Change of File Checkbox
61     If Not Intersect(Target, Range("Q9:Q1000" )) Is Nothing And Target.Value <> Empty Then
62         If Target.Value = Sheet2.Range("UncheckIcon" ).Value Then Target.Value = Sheet2.Range(
63             "CheckIcon" ).Value Else: Target.Value = Sheet2.Range("UncheckIcon" ).Value

```

```
54 1 2 Range("A1" ).Select 'Unselect the current cell
55 End If
56 'On Selection change of any field
57 If Not Intersect(Target, Range("R9:W1000" )) Is Nothing And Target.Value <> Empty Then
58 Range("Q9:Q" & Range("Q9999" ).End(xlUp).Row).Value = Sheet2.Range("UncheckIcon" ).
Value
59 Range("B1" ).Value = Target.Row
60 Range("Q" & Target.Row).Value = Sheet2.Range("CheckIcon" )
61 End If
62
63 'On Selection Change of any column header
64 If Not Intersect(Target, Range("S6:W6" )) Is Nothing Then FileSort
65
66 If Not Intersect(Target, Range("Z11:Z15" )) Is Nothing Then ShowHideColumns
67 End Sub
68
```

A

Application, 2

C

Cells, 2

CloseFolder, 2

CountLarge, 2

CreateObject, 2

E

Empty, 2, 3

ExpandFolder, 2

Explicit, 2

F

FileOpenMsg, 2

FileSort, 3

Filter_Run, 2

G

GetFolder, 2

I

InStr, 2

Intersect, 2, 3

M

MoveFile, 2

MsgBox, 2

N

Name, 2

NewFolderName, 2

O

objFolder, 2

objFSO, 2

Offset, 2

OldFolderName, 2

OpenFolder, 2

P

ParentFolder, 2

R

Range, 2, 3

Row, 2, 3

S

SendKeys, 2

Sheet2, 2, 3

ShowHideColumns, 3

T

Target, 2, 3

V

Value, 2, 3

W

Worksheet_Change, 2

Worksheet_SelectionChange, 2

X

xUp, 3

1 Option Explicit

2

E

Explicit, 5

1 Option Explicit

2

E

Explicit, 7

1 Option Explicit
2

E

Explicit, 9

```
1 Option Explicit
2 Dim StartTime, SecondsElapsed As Double
3
4 Sub ResetCalc() 'This macro reset Calculations, screen updating and events and should be
   used at the end of a macro when StopCalc has been used
5     With Application
6         .EnableEvents = True 'Re-enabled Events (optional)
7         .Calculation = xlCalculationAutomatic 'Resets workbook Calculations to Automatic
8         .ScreenUpdating = True 'Turns on Screen Updating
9     End With
10 End Sub
11
12
13
14 'Start Clock and StopClock Macros can be used inside other macros to set a Time on Macros.
   This helps test the speed of macros, so when you make changes, you can see how it affects
   the speed of those macros
15
16 Sub StartClock() 'Starts the Timer
17     StartTime = Timer 'Starts Timer
18 End Sub
19
20 Sub StopCalc() 'This macro stops calculations and updating. Disabling events can be
   included as well. By adding this macro, at the beginning of a Macro then using the ResetCalc,
   your code is sped up considerably
21     With Application
22         .EnableEvents = False 'Turns Off Enabling Events '(optional)
23         .Calculation = xlCalculationManual 'Sets Workbook Calculation to Manual
24         .ScreenUpdating = False 'Turns Off Screen Updating
25     End With
26 End Sub
27
28 Sub StopClock()
29     SecondsElapsed = Round(Timer - StartTime, 2) 'Determine how many seconds code took to run
30     MsgBox "This code ran successfully in " & SecondsElapsed & " seconds" , vbInformation
   'Notifies the user how long the macro took to run (in seconds)
31 End Sub
32
33
```

A

Application, 11

C

Calculation, 11

E

Explicit, 11

M

MsgBox, 11

R

ResetCalc, 11

Round, 11

S

ScreenUpdating, 11

SecondsElapsed, 11

StartClock, 11

StartTime, 11

StopCalc, 11

StopClock, 11

T

Timer, 11

V

vbInformation, 11

X

xlCalculationAutomatic, 11

xlCalculationManual, 11

```
1 Option Explicit
2
3 Sub DisplayDrives()
4 Dim objFSO, objDrive, objName As Object
5 Dim RowNum As Long
6 Set objFSO = CreateObject("Scripting.FileSystemObject")
7 With Sheet1
8 StopCalc
9 .Range("B2").Value = True 'Set Folder Load to True
10 .Range("A9:B9999,D9:O9999,Q9:Q9999,S9:X9999").ClearContents 'Clear Existing Files &
    Folders
11 RowNum = 9
12 For Each objDrive In objFSO.Drives
13 Set objName = objFSO.GetDrive(objFSO.GetDriveName(objDrive))
14 On Error GoTo NextDrive
15 .Range("B" & RowNum).Value = objDrive.Path & "\"
16 .Range("D" & RowNum).Value = Sheet2.Range("DriveIcon").Value
17 On Error Resume Next
18 .Range("E" & RowNum).Value = Format((objDrive.TotalSize - objDrive.FreeSpace) /
    objDrive.TotalSize, "percent") 'Show Percent Full
19 .Range("E" & RowNum & ":H" & RowNum).Merge
20 .Range("I" & RowNum).Value = objDrive.VolumeName & " (" & objDrive.DriveLetter &
    ":)"
21 RowNum = RowNum + 1
22 NextDrive:
23 Next objDrive
24 .Range("B5,H5").Value = "My Computer"
25 .Range("G5").Value = Sheet2.Range("MyComputer").Value
26 ResetCalc
27 .Range("B2").Value = False 'Set Folder Load to False
28 .Range("A1").Select
29 End With
30 End Sub
31
```


C

ClearContents, [13](#)
CreateObject, [13](#)

D

DisplayDrives, [13](#)
DriveLetter, [13](#)
Drives, [13](#)

E

Explicit, [13](#)

F

Format, [13](#)
FreeSpace, [13](#)

G

GetDrive, [13](#)
GetDriveName, [13](#)

M

Merge, [13](#)

N

NextDrive, [13](#)

O

objDrive, [13](#)
objFSO, [13](#)
objName, [13](#)

P

Path, [13](#)

R

Range, [13](#)
ResetCalc, [13](#)
RowNum, [13](#)

S

Sheet1, [13](#)
Sheet2, [13](#)
StopCalc, [13](#)

T

TotalSize, [13](#)

V

Value, [13](#)
VolumeName, [13](#)

```
1  Option Explicit
2
3
4  Sub FadeDefaultMess()
5      Dim i As Integer
6      Dim delay As Double
7      Dim StartTime As Double
8      With ActiveSheet.Shapes("SetDefaultMsg" )
9          .Visible = True
10         For i = 1 To 100
11             .Fill.Transparency = i / 100
12             delay = 0.006
13             StartTime = Timer
14             Do
15                 DoEvents
16                 Loop While Timer - StartTime < delay
17             Next i
18             .Visible = False
19         End With
20     End Sub
21
22
23
24
```

A

ActiveSheet, 15

D

delay, 15

DoEvents, 15

E

Explicit, 15

F

Fill, 15

FoldDefaultMess, 15

I

i, 15

S

Shapes, 15

StartTime, 15

T

Timer, 15

Transparency, 15

V

Visible, 15

```

1 Option Explicit
2 Dim LastRow, FileRow, LastFiltRow As Long
3 Dim Resp As Integer
4
5 Sub CopySelectedFiles()
6     With Sheet1
7         LastRow = .Range("S9999").End(xlUp).Row
8         If LastRow < 9 Then Exit Sub
9         .Range("EA3:EH999").ClearContents 'Clear any remaining Results
10        .Range("Q8:X" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("DA2:DH3"
11        ), CopyToRange:=.Range("EA2:EH2"), Unique:=True
12    End With
13 End Sub
14
15 Sub DeleteSelectedFiles()
16     Dim DeleteFilename As String
17     Resp = MsgBox("Are you sure you want to delete the selected files?", vbYesNo, "Delete
18     Files" )
19     If Resp = vbNo Then Exit Sub
20     With Sheet1
21         LastRow = .Range("S9999").End(xlUp).Row
22         If LastRow < 9 Then Exit Sub
23         .Range("BA3:BH99,CA3:CH999").ClearContents 'Clear any remaining Criteria Or Results
24         .Range("BA3").Value = Sheet2.Range("CheckIcon").Value
25         .Range("Q8:X" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("BA2:BH3"
26         ), CopyToRange:=.Range("CA2:CH2"), Unique:=True
27         LastFiltRow = .Range("CH99999").End(xlUp).Row
28         If LastFiltRow < 3 Then
29             MsgBox "Please select files to delete first"
30             Exit Sub
31         End If
32         StopCalc
33         For FileRow = 3 To LastFiltRow
34             DeleteFilename = .Range("CH" & FileRow).Value 'Determine Source Filename
35             'On Error Resume Next
36             Kill (DeleteFilename)
37             'On Error GoTo 0
38             NextFile:
39         Next FileRow
40     End With
41     ReloadFolder
42     ResetCalc
43 End Sub
44
45 Sub PasteSelectedFiles()
46     Dim SourceFilename, DestinationFilename As String
47     With Sheet1
48         LastRow = .Range("EC99999").End(xlUp).Row
49         If LastRow < 3 Then
50             MsgBox "No files have been copied to the clipboard to paste. Please select and copy
51             files to be pasted"
52             Exit Sub
53         End If
54         StopCalc
55         For FileRow = 3 To LastRow
56             SourceFilename = .Range("EH" & FileRow).Value 'Determine Source Filename
57             DestinationFilename = .Range("B5").Value & "\" & .Range("EC" & FileRow).Value
58             'Determine Destination Filename
59             'On Error Resume Next

```

```

1 2 3
57 If Dir(DestinationFilename) <> "" Then 'File already exists
58 'On Error GoTo 0
59 Resp = MsgBox("The file " & DestinationFilename & vbCrLf & "already exists.
        Would you like to overwrite this file?" , vbYesNo, "File Exists" )
60 If Resp = vbNo Then GoTo NextFile
61 End If
62 'Two Different Copy Methods, use just one
63 FileCopy SourceFilename, DestinationFilename 'Simple File Copy Method
64 'Call ObjFile.CopyFile(SourceFilename, DestinationFilename, True) 'Using FSO Option
        -True means Overwrite, False means don't overwrite
65 NextFile:
66 Next FileRow
67 ReloadFolder 'Reload Current Folder
68 End With
69 ResetCalc
70 End Sub

71
72
73 Sub ReloadFolder()
74 Dim ObjFile, objFSO, objFolder As Object
75 Set objFSO = VBA.CreateObject("Scripting.FileSystemObject" )
76
77 With Sheet1
78 .Range("B2" ).Value = True 'Seat Load Files True
79 .Range("Q9:Q9999,S9:X9999" ).ClearContents 'Clear Existing Folder to reload
80 FileRow = 9
81 Set objFolder = objFSO.GetFolder(.Range("B5" ).Value)
82 For Each ObjFile In objFolder.Files
83 .Range("Q" & FileRow).Value = Sheet2.Range("UncheckIcon" ).Value
84 .Range("S" & FileRow).Value = ObjFile.Name
85 .Range("T" & FileRow).Value = ObjFile.Type
86 .Range("U" & FileRow).Value = ObjFile.Size
87 .Range("V" & FileRow).Value = ObjFile.DateCreated
88 .Range("W" & FileRow).Value = ObjFile.DateLastModified
89 .Range("X" & FileRow).Value = ObjFile.Path
90 FileRow = FileRow + 1
91 Next ObjFile
92 .Range("A1" ).Select
93 .Range("B2" ).Value = False 'Set Load Files To False
94 ActiveWindow.ScrollRow = 9
95 End With
96 End Sub

97
98
99
100 Sub SelectedFilesFilter()
101 With Sheet1
102 LastRow = .Range("S9999" ).End(xlUp).Row
103 If LastRow < 9 Then Exit Sub
104 .Range("BA3:BH99,CA3:CH999" ).ClearContents 'Clear any remaining Criteria Or Results
105 .Range("BA3" ).Value = Sheet2.Range("CheckIcon" ).Value
106 .Range("Q7:X" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("BA2:BH3"
        ), CopyToRange:=.Range("CA2:CH2" ), Unique:=True
107 End With
108 End Sub

```

A

ActiveWindow, 18
AdvancedFilter, 17, 18

C

ClearContents, 17, 18
CopySelectedFiles, 17
CopyToRange, 17, 18
CreateObject, 18
CriteriaRange, 17, 18

D

DateCreated, 18
DateLastModified, 18
DeleteFilename, 17
DeleteSelectedFiles, 17
DestinationFilename, 17, 18
Dir, 18

E

Explicit, 17

F

FileCopy, 18
FileRow, 17, 18
Files, 18

G

GetFolder, 18

K

Kill, 17

L

LastFiltRow, 17
LastRow, 17, 18

M

MsgBox, 17, 18

N

Name, 18
NextFile, 17, 18

O

ObjFile, 18
objFolder, 18
objFSO, 18

P

PasteSelectedFiles, 17
Path, 18

R

Range, 17, 18
ReloadFolder, 17, 18
ResetCalc, 17, 18
Resp, 17, 18
Row, 17, 18

S

ScrollRow, 18
SelectedFilesFilter, 18
Sheet1, 17, 18
Sheet2, 17, 18
Size, 18
SourceFilename, 17, 18
StopCalc, 17

U

Unique, 17, 18

V

Value, 17, 18
VBA, 18
vbCrLf, 18
vbNo, 17, 18
vbYesNo, 17, 18

X

xlFilterCopy, 17, 18
xlUp, 17, 18

Option Explicit

```

1  Option Explicit
2
3
4  Sub Filter_Clear()
5      StopCalc
6      With Sheet1
7          .Range("S7:W7").Value = .Range("AM7:AQ7").Value 'Reset Default Search Text
8          .Range("BA3:BH3,CA3:CA9999").ClearContents 'Clear any Previous Search Criteria or
          Results
9          .Shapes("ClearFiltBtn").Visible = msoFalse
10     End With
11     ReloadFolder
12     ResetCalc
13 End Sub
14
15
16 Sub Filter_Run()
17     Dim LastRow, LastFiltRow As Long
18     StopCalc
19     With Sheet1
20         ReloadFolder
21         .Range("B2").Value = True 'Set File Load To True
22         LastRow = .Range("S9999").End(xlUp).Row
23         .Range("CA3:CA9999").ClearContents
24         If InStr(.Range("S7").Value, ":") <> 0 Or .Range("S7").Value = Empty Then .Range(
           "BC3").ClearContents Else .Range("BC3").Value = "*" & .Range("S7").Value & "*"
           'Either Clear or Set Name Filter
25         If InStr(.Range("T7").Value, ":") <> 0 Or .Range("T7").Value = Empty Then .Range(
           "BD3").ClearContents Else .Range("BD3").Value = "*" & .Range("T7").Value & "*"
           'Either Clear or Set File Type Filter
26         If InStr(.Range("U7").Value, ":") <> 0 Or .Range("U7").Value = Empty Then .Range(
           "BE3").ClearContents Else .Range("BE3").Value = .Range("U7").Value 'Either Clear or
           Set File Size
27         If InStr(.Range("V7").Value, ":") <> 0 Or .Range("V7").Value = Empty Then .Range(
           "BF3").ClearContents Else .Range("BF3").Value = .Range("V7").Value 'Either Clear
           or Set Date Created
28         If InStr(.Range("W7").Value, ":") <> 0 Or .Range("W7").Value = Empty Then .Range(
           "BG3").ClearContents Else .Range("BG3").Value = .Range("W7").Value 'Either Clear
           or Set Date Modified
29         .Range("Q8:X" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range("BA2:BH3"
           ), CopyToRange:=.Range("CA2:CH2"), Unique:=True
30         LastFiltRow = .Range("CC9999").End(xlUp).Row
31         .Range("Q9:Q9999,S9:X9999").ClearContents
32         .Shapes("ClearFiltBtn").Visible = msoCTrue
33         If LastFiltRow < 3 Then
34             .Range("B2").Value = False 'Set File Load To False
35             ResetCalc
36             Exit Sub
37         End If
38         .Range("S9:X" & LastFiltRow + 6).Value = .Range("CC3:CH" & LastFiltRow).Value
39         .Range("B2").Value = False 'Set File Load To False
40     End With
41     ResetCalc
42 End Sub

```

A

AdvancedFilter, [20](#)

C

ClearContents, [20](#)

CopyToRange, [20](#)

CriteriaRange, [20](#)

E

Empty, [20](#)

Explicit, [20](#)

F

Filter_Clear, [20](#)

Filter_Run, [20](#)

I

InStr, [20](#)

L

LastFiltRow, [20](#)

LastRow, [20](#)

M

msoCTrue, [20](#)

msoFalse, [20](#)

R

Range, [20](#)

ReloadFolder, [20](#)

ResetCalc, [20](#)

Row, [20](#)

S

Shapes, [20](#)

Sheet1, [20](#)

StopCalc, [20](#)

U

Unique, [20](#)

V

Value, [20](#)

Visible, [20](#)

X

xlFilterCopy, [20](#)

xlUp, [20](#)


```

1
2 Option Explicit
3 Dim objFSO, objFolder, ObjSubFolder, ObjFile As Object
4 Dim Rng, FoundLast As Range
5
6
7 Sub CloseFolder()
8     Dim SubfolderQty, RowNum, ItemNum, LastRow, ColNum, NextRow, FileRow, ActRow, FoldRow As Long
9     Set objFSO = CreateObject("Scripting.FileSystemObject")
10    StopCalc
11    With Sheet1
12        Set Rng = Sheet1.Range("D9:O1000")
13        Set FoundLast = Rng.Find(What:="*", After:=Rng.Cells(1), Lookat:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByRows, SearchDirection:=xlPrevious, MatchCase:=False)
14        LastRow = FoundLast.Row
15        .Range("B2").Value = True 'Set Load Files True
16        ActiveCell.Value = Sheet2.Range("CloseFold").Value
17        RowNum = ActiveCell.Row
18        ColNum = ActiveCell.Column
19        If ActiveCell.Offset(1, 0).Value <> Empty Then GoTo SkipCopyPaste
20        NextRow = ActiveCell.Offset(0, 0).End(xlDown).Row - 1
21        If NextRow > 9999 Then 'Closing Last Folder
22            .Range("B" & RowNum + 1 & ":B" & LastRow).ClearContents
23            .Range("D" & RowNum + 1 & ":O" & LastRow).ClearContents
24            GoTo SkipCopyPaste
25        End If
26        .Range("D" & NextRow + 1 & ":O" & LastRow).Copy
27        .Range("D" & RowNum + 1).PasteSpecial xlPasteValues
28        .Range("B" & NextRow + 1 & ":B" & LastRow).Copy
29        .Range("B" & RowNum + 1).PasteSpecial xlPasteValues
30        .Range("D" & LastRow - NextRow + RowNum + 1 & ":O" & LastRow).ClearContents
31        .Range("B" & LastRow - NextRow + RowNum + 1 & ":B" & LastRow).ClearContents
32        SkipCopyPaste:
33        FileRow = 9
34        .Range("Q9:Q9999,S9:X9999").ClearContents 'Clear Existing Folder to reload
35        On Error GoTo NoPath
36        Set objFolder = objFSO.GetFolder(.Range("B" & RowNum).Value)
37        On Error Resume Next
38        For Each ObjFile In objFolder.Files
39            .Range("Q" & FileRow).Value = Sheet2.Range("UncheckIcon").Value
40            .Range("S" & FileRow).Value = ObjFile.Name
41            .Range("T" & FileRow).Value = ObjFile.Type
42            .Range("U" & FileRow).Value = ObjFile.Size
43            .Range("V" & FileRow).Value = ObjFile.DateCreated
44            .Range("W" & FileRow).Value = ObjFile.DateLastModified
45            .Range("X" & FileRow).Value = ObjFile.Path
46            FileRow = FileRow + 1
47        Next ObjFile
48        .Range("A1").Select
49        .Range("B5").Value = .Range("B" & RowNum).Value 'Set Current Folder
50        ActiveWindow.ScrollRow = 9
51        ResetCalc
52        .Range("H5").Value = Replace(.Range("B5").Value, "\", " > ") 'FilePath
53        .Range("B2").Value = False 'Set Load Files To False
54        .Range("A1").Select
55    End
56
57    ResetCalc
58    Exit Sub
59    NoPath:

```

```

1 2
60 .Range("B2" ).Value = False 'Set Load Files To False
61 ResetCalc
62 'MsgBox "This folder may have been moved or deleted. Please refresh the folder list"
63 End With
64 End Sub
65
66 Sub CreateNewFolder()
67 Dim ParentFolder As String
68 Dim FoldRng, FindParentRng As Range
69 Dim SubfolderQty, RowNum, ItemNum, LastRow, ColNum, NextRow, FileRow, ActRow, FoldRow As
Long
70 With Sheet1
71 .Range("B2" ).Value = True 'Set Folder Load to True
72 Set objFSO = CreateObject("Scripting.FileSystemObject" )
73 Set FoldRng = .Range("B8:B" & .Range("B99999" ).End(xlUp).Row) 'Set Current Folder
Range
74 ParentFolder = .Range("B5" ).Value
75 On Error GoTo NoPath
76 Set objFolder = objFSO.CreateFolder(ParentFolder & "\New Folder" )
77 Set FindParentRng = FoldRng.Find(ParentFolder, , xlValues, xlWhole)
78 Set Rng = .Range("D9:O999" )
79 Set FoundLast = Rng.Find(What:="*", After:=Rng.Cells(1), Lookat:=xlPart, LookIn:=
xlFormulas, SearchOrder:=xlByRows, SearchDirection:=xlPrevious, MatchCase:=False)
80 LastRow = FoundLast.Row
81 If Not FindParentRng Is Nothing Then
82 FoldRow = FindParentRng.Row
83 .Range("D" & FoldRow + 1 & ":O" & LastRow).Copy
84 .Range("D" & FoldRow + 2).PasteSpecial xlPasteValues
85 .Range("A" & FoldRow + 1 & ":B" & LastRow).Copy
86 .Range("A" & FoldRow + 2).PasteSpecial xlPasteValues
87 Application.CutCopyMode = False
88 Else:
89 FoldRow = LastRow 'Already At Top Level, Add New Folder on Last Row
90 End If
91 .Range("B2" ).Value = False 'Set Folder Load to False
92 .Range("B" & FoldRow + 1).Value = ParentFolder & "\New Folder"
93 .Range("A" & FoldRow + 1).Value = False 'New Folder contains no subfolders
94 ColNum = .Range("O" & FoldRow + 1).End(xlToLeft).Column
95 If ColNum < 5 Then ColNum = 5
96 If ColNum <> 5 Then .Cells(FoldRow + 1, ColNum - 1).ClearContents 'clear any icons
97 .Cells(FoldRow + 1, ColNum - 1).Value = Sheet2.Range("CloseFold" ).Value 'Add Folder
icon is necessary
98 .Cells(FoldRow + 1, ColNum).Value = "New Folder"
99 .Cells(FoldRow + 1, ColNum).Select
100 Application.Wait Now + 0.00001
101 Application.SendKeys "{BACKSPACE}"
102 Application.SendKeys "^{z}"
103 Exit Sub
104 NoPath:
105 .Range("B2" ).Value = False 'Set Folder Load to False
106 MsgBox "Please check that 'New Folder' does not already exist, or select a drive or
folder to add a new folder"
107 End With
108 End Sub
109
110
111 Sub ExpandFolder()
112 Dim SubfolderQty, RowNum, ItemNum, LastRow, ColNum, NextRow, FileRow, ActRow, FoldRow As
Long
113 Set objFSO = CreateObject("Scripting.FileSystemObject" )
114 StopCalc

```

```

1
115 With Sheet1
116     Set Rng = .Range("D9:O1000" )
117     Set FoundLast = Rng.Find(What:="*" , After:=Rng.Cells(1), Lookat:=xlPart, LookIn:=
xlFormulas, SearchOrder:=xlByRows, SearchDirection:=xlPrevious, MatchCase:=False)
118     .Range("B2" ).Value = True 'Set Load Files True
119     .Range("Q9:Q9999,S9:X9999" ).ClearContents 'Clear Files
120     If ActiveCell.Value <> Sheet2.Range("DriveIcon" ).Value Then ActiveCell.Value = Sheet2.
Range("OpenFold" ).Value
121     RowNum = ActiveCell.Row
122     ActRow = ActiveCell.Row
123     ColNum = ActiveCell.Column + 1
124     LastRow = FoundLast.Row
125     Sheet1.Range("B3" ).Value = .Range("B5" ).Value 'Set Previous Folder
126     .Range("B5" ).Value = .Range("B" & ActiveCell.Row).Value
127     On Error GoTo NoPath
128     Set objFolder = objFSO.GetFolder(.Range("B5" ).Value)
129
130     Set ObjSubFolder = objFolder.Subfolders
131     On Error Resume Next
132     SubfolderQty = ObjSubFolder.Count
133     If Err.Number <> 0 Then
134         'On Error GoTo SkipFolder
135
136         For Each ObjSubFolder In objFolder.Subfolders
137             SubfolderQty = SubfolderQty + 1
138         Next ObjSubFolder
139     End If
140     'On Error GoTo 0
141     If SubfolderQty < 1 Then GoTo SkipExpand
142     .Range("D" & RowNum + 1 & ":O" & LastRow).Copy
143     .Range("D" & RowNum + 1 + SubfolderQty & ":O" & LastRow + SubfolderQty).PasteSpecial
xlPasteValues
144     .Range("A" & RowNum + 1 & ":B" & LastRow).Copy
145     .Range("A" & RowNum + 1 + SubfolderQty & ":B" & LastRow + SubfolderQty).PasteSpecial
xlPasteValues
146     Application.CutCopyMode = False
147     Cells(RowNum, ColNum).Select
148     .Range("D" & RowNum + 1 & ":O" & RowNum + SubfolderQty).ClearContents
149
150     For Each ObjSubFolder In objFolder.Subfolders
151         .Cells(RowNum + 1, ColNum).Value = Sheet2.Range("CloseFold" ).Value
152         .Cells(RowNum + 1, ColNum + 1).Value = ObjSubFolder.Name
153         .Range("B" & RowNum + 1).Value = ObjSubFolder.Path
154         If ObjSubFolder.Subfolders.Count = 0 Then .Range("A" & RowNum + 1).Value = False
Else: .Range("A" & RowNum + 1).Value = True
155         RowNum = RowNum + 1
156     Next ObjSubFolder
157
158     SkipExpand:
159     FileRow = 9
160     Set objFolder = objFSO.GetFolder(.Range("B" & ActRow).Value)
161     For Each ObjFile In objFolder.Files
162         .Range("Q" & FileRow).Value = Sheet2.Range("UncheckIcon" ).Value
163         .Range("S" & FileRow).Value = ObjFile.Name
164         .Range("T" & FileRow).Value = ObjFile.Type
165         .Range("U" & FileRow).Value = ObjFile.Size
166         .Range("V" & FileRow).Value = ObjFile.DateCreated
167         .Range("W" & FileRow).Value = ObjFile.DateLastModified
168         .Range("X" & FileRow).Value = ObjFile.Path
169         FileRow = FileRow + 1
170     Next ObjFile

```

1 2

```

1 2
171 .Range("B5" ).Value = .Range("B" & ActRow).Value 'Set Current Folder
172 .Range("H5" ).Value = Replace(.Range("B5" ).Value, "\", " > " ) 'FilePath
173 .Range("A1" ).Select
174 .Range("B2" ).Value = False 'Set Load Files To False
175 ActiveWindow.ScrollRow = 9
176
177 ResetCalc
178 End
179 Exit Sub
180 NoPath:
181 .Range("B2" ).Value = False
182 ResetCalc
183 MsgBox "This folder may have been moved or deleted. Please refresh the folder list"
184 End With
185 End Sub
186 Sub GoToHomeFolder()
187 Sheet1.Range("B3" ).Value = Sheet1.Range("B5" ).Value 'Set Previous Folder
188 Sheet1.Range("B5" ).Value = Sheet1.Range("B6" ).Value 'Set Home Folder as Current folder
189
190 OpenFolder
191 End Sub
192 Sub GoToNextFolder()
193 Sheet1.Range("B5" ).Value = Sheet1.Range("B3" ).Value 'Set Current folder
194 Sheet1.Range("B4" ).Value = Sheet1.Range("B3" ).Value 'Set Next Folder
195 Sheet1.Range("B3" ).Value = Sheet1.Range("B5" ).Value 'Set Previous Folder
196 OpenFolder
197 End Sub
198 Sub GoToParentFolder()
199 Set objFSO = CreateObject("Scripting.FileSystemObject" )
200 On Error GoTo HighestLevel
201 Set objFolder = objFSO.GetFolder(Sheet1.Range("B5" ).Value)
202 Sheet1.Range("B3" ).Value = Sheet1.Range("B5" ).Value 'Set Previous Folder
203 On Error GoTo DriveLoc
204 Sheet1.Range("B5" ).Value = objFolder.ParentFolder.Path
205 OpenFolder
206 Exit Sub
207 DriveLoc:
208 DisplayDrives
209 Exit Sub
210 HighestLevel:
211 MsgBox "You are at the the drive level"
212 End Sub
213 Sub GoToPrevFolder()
214 If Sheet1.Range("B3" ).Value = Empty Then
215 MsgBox "You are at the last record" 'Future Versions of this applicaiton save a
216 history of these folders
217 Exit Sub
218 End If
219 Sheet1.Range("B5" ).Value = Sheet1.Range("B3" ).Value 'Set Current folder
220 Sheet1.Range("B4" ).Value = Sheet1.Range("B3" ).Value 'Set Next Folder
221 OpenFolder
222 End Sub
223 Sub OpenFolder()
224 Dim SubfolderQty, RowNum, ItemNum, LastRow, ColNum, NextRow, FileRow, ActRow, FoldRow As
225 Long
226 StopCalc
227 Set objFSO = CreateObject("Scripting.FileSystemObject" )
228 With Sheet1
229 .Range("B2" ).Value = True 'Set Folder Load to True
230 Set objFolder = objFSO.GetFolder(.Range("B5" ).Value)

```

1 2

```
1 2
229 Set ObjSubFolder = objFolder.Subfolders
230 On Error Resume Next
231 SubfolderQty = ObjSubFolder.Count
232 .Range("G5").Value = Sheet2.Range("CloseFold").Value 'Set Icon
233 RowNum = 9
234 .Range("A9:B9999,D9:09999,Q9:Q9999,S9:X9999").ClearContents
235 .Range("E9:H99").UnMerge
236 .Range("FD9:FP1000").Copy 'Added in Part 2
237 .Range("D9").PasteSpecial xlPasteFormats 'Added in Part 2
238 Application.CutCopyMode = False 'Added in Part 2
239 Sheet1.Range("B3").Value = Sheet1.Range("B5").Value 'Set Previous Folder
240 For Each ObjSubFolder In objFolder.Subfolders
241     .Range("D" & RowNum).Value = Sheet2.Range("CloseFold").Value
242     .Range("E" & RowNum).Value = ObjSubFolder.Name
243     .Range("B" & RowNum).Value = ObjSubFolder.Path
244     If ObjSubFolder.Subfolders.Count = 0 Then .Range("A" & RowNum).Value = False Else:
245         .Range("A" & RowNum).Value = True
246     RowNum = RowNum + 1
247 Next ObjSubFolder
248 FileRow = 9
249 For Each ObjFile In objFolder.Files
250     .Range("X" & FileRow).Value = ObjFile.Path
251     .Range("Q" & FileRow).Value = Sheet2.Range("UncheckIcon").Value
252     .Range("S" & FileRow).Value = ObjFile.Name
253     .Range("T" & FileRow).Value = ObjFile.Type
254     .Range("U" & FileRow).Value = ObjFile.Size
255     .Range("V" & FileRow).Value = ObjFile.DateCreated
256     .Range("W" & FileRow).Value = ObjFile.DateLastModified
257     FileRow = FileRow + 1
258 Next ObjFile
259 .Range("H5").Value = Replace(.Range("B5").Value, "\", ">" ) 'FilePath
260 ActiveWindow.ScrollRow = 9
261 .Range("B2").Value = False 'Set Load Files To False
262 .Range("A1").Select
263 End With
264 ResetCalc
265 End Sub
266
267 Sub SetAsDefaultFolder()
268     Sheet1.Range("B6").Value = Sheet1.Range("B5").Value 'Set Default Folder
269     FoldDefaultMess
270 End Sub
```

A

ActiveCell, 22, 24
 ActiveWindow, 22, 25, 26
 ActRow, 22-25
 After, 22-24
 Application, 23, 24, 26

C

Cells, 22-24
 ClearContents, 22-24, 26
 CloseFolder, 22
 ColNum, 22-25
 Column, 22-24
 Copy, 22-24, 26
 Count, 24, 26
 CreateFolder, 23
 CreateNewFolder, 23
 CreateObject, 22, 23, 25
 CutCopyMode, 23, 24, 26

D

DateCreated, 22, 24, 26
 DateLastModified, 22, 24, 26
 DisplayDrives, 25
 DriveLoc, 25

E

Empty, 22, 25
 Err, 24
 ExpandFolder, 23
 Explicit, 22

F

FileRow, 22-26
 Files, 22, 24, 26
 Find, 22-24
 FindParentRng, 23
 FoldDefaultMess, 26
 FoldRng, 23
 FoldRow, 22, 23, 25
 FoundLast, 22-24

G

GetFolder, 22, 24, 25
 GoToHomeFolder, 25
 GoToNextFolder, 25
 GoToParentFolder, 25
 GoToPrevFolder, 25

H

HighestLevel, 25

I

ItemNum, 22, 23, 25

L

LastRow, 22-25
 Lookat, 22-24
 LookIn, 22-24

M

MatchCase, 22-24
 MsgBox, 23, 25

N

Name, 22, 24, 26
 NextRow, 22, 23, 25
 NoPath, 22-25
 Now, 23

Number, 24

O

ObjFile, 22, 24, 26
 objFolder, 22-26
 objFSO, 22-25
 ObjSubFolder, 22, 24, 26
 Offset, 22
 OpenFolder, 25

P

ParentFolder, 23, 25
 PasteSpecial, 22-24, 26
 Path, 22, 24-26

R

Range, 22-26
 Replace, 22, 25, 26
 ResetCalc, 22, 23, 25, 26
 Rng, 22-24
 Row, 22-24
 RowNum, 22-26

S

ScrollRow, 22, 25, 26
 SearchDirection, 22-24
 SearchOrder, 22-24
 SendKeys, 23
 SetAsDefaultFolder, 26
 Sheet1, 22-26
 Sheet2, 22-24, 26
 Size, 22, 24, 26
 SkipCopyPaste, 22
 SkipExpand, 24
 StopCalc, 22, 23, 25
 SubfolderQty, 22-26
 Subfolders, 24, 26

U

UnMerge, 26

V

Value, 22-26

W

Wait, 23
 What, 22-24

X

xlByRows, 22-24
 xlDown, 22
 xlFormulas, 22-24
 xlPart, 22-24
 xlPasteFormats, 26
 xlPasteValues, 22-24
 xlPrevious, 22-24
 xlToLeft, 23
 xlUp, 23
 xlValues, 23
 xlWhole, 23

```
1 Option Explicit
2
3 Sub HideOptions()
4     With Sheet1
5         .Range("Z:AA").EntireColumn.Hidden = True
6         .Shapes("HideOptBtn").Visible = msoFalse
7         .Shapes("ShowOptBtn").Visible = msoCTrue
8     End With
9 End Sub
10
11
12 Sub ShowHideColumns()
13     If ActiveCell.Value = Sheet2.Range("CheckIcon").Value Then 'Hide Column
14         ActiveCell.Value = Sheet2.Range("UncheckIcon").Value
15         Sheet1.Cells(1, ActiveCell.Row + 9).EntireColumn.Hidden = True
16     Else: 'Show Column
17         ActiveCell.Value = Sheet2.Range("CheckIcon").Value
18         Sheet1.Cells(1, ActiveCell.Row + 9).EntireColumn.Hidden = False
19     End If
20     Sheet1.Range("A1").Select
21 End Sub
22
23 Sub ShowOptions()
24     With Sheet1
25         .Range("Z:AA").EntireColumn.Hidden = False
26         .Shapes("HideOptBtn").Visible = msoCTrue
27         .Shapes("ShowOptBtn").Visible = msoFalse
28     End With
29 End Sub
```

A

ActiveCell, [28](#)

C

Cells, [28](#)

E

EntireColumn, [28](#)

Explicit, [28](#)

H

Hidden, [28](#)

HideOptions, [28](#)

M

msoCTrue, [28](#)

msoFalse, [28](#)

R

Range, [28](#)

Row, [28](#)

S

Shapes, [28](#)

Sheet1, [28](#)

Sheet2, [28](#)

ShowHideColumns, [28](#)

ShowOptions, [28](#)

V

Value, [28](#)

Visible, [28](#)


```

1 Option Explicit
2
3 Sub FileSort()
4 Dim ActCol, LastRow, HeadCol As Long
5 Dim ActHeader, SortType As String 'Dimension String variables
6 With Sheet1
7 .Range("B2").Value = True 'Set File Load To True
8 StopCalc 'Run Macro to Stop Calculation & Screen Updating
9 ActHeader = ActiveCell.Value 'Determine Active Header Value for sorting
10 LastRow = .Range("S9999").End(xlUp).Row 'Get the last Row of the displayed table
11 .Sort.SortFields.Clear 'Clear any previous sorts on the table.
12 If InStr(ActHeader, Sheet2.Range("SortAlphaAZ").Value) <> 0 Or InStr(ActHeader, Sheet2
13 .Range("SortNum19").Value) <> 0 Then SortType = "Desc" 'Check if the active column
14 is set for Ascending and if so, then set Sort Type to Descending
15 If InStr(ActHeader, Sheet2.Range("SortAlphaZA").Value) <> 0 Or InStr(ActHeader, Sheet2
16 .Range("SortNum91").Value) <> 0 Then SortType = "Asc" 'Check if the active column is
17 set for Descending and if so, then set Sort Type to Ascending
18 For HeadCol = 19 To 23 'Set loop for each column header to remove any Ascending or
19 Descending
20 .Cells(6, HeadCol).Value = Replace(Replace(Replace(Replace(.Cells(6, HeadCol).Value,
21 " " & Sheet2.Range("SortAlphaAZ").Value, "" ), " " & Sheet2.Range("SortAlphaZA"
22 ).Value, "" ), " " & Sheet2.Range("SortNum19").Value, "" ), " " & Sheet2.Range(
23 "SortNum91").Value, "" ) 'Remove any Ascending or Descending Alpha or Numerical
24 Icons
25 Next HeadCol
26 'Run Sorting
27 If SortType = "Desc" Then 'If the sort type is Descending
28 .Sort.SortFields.Add Key:=.Cells(9, ActiveCell.Column), SortOn:=xlSortOnValues,
29 Order:=xlDescending, DataOption:=xlSortNormal 'Set Sort Parameters for Descending
30 Sort
31 If ActiveCell.Column < 21 Then ActiveCell.Value = ActiveCell.Value & " " & Sheet2.
32 Range("SortAlphaZA").Value Else: ActiveCell.Value = ActiveCell.Value & " " &
33 Sheet2.Range("SortNum91").Value 'Add in Descending Numerical or Alpha Character
34 Else: 'Sort type is Ascending
35 .Sort.SortFields.Add Key:=.Cells(9, ActiveCell.Column), SortOn:=xlSortOnValues,
36 Order:=xlAscending, DataOption:=xlSortNormal 'Set Sort Parameters for Ascending
37 Sort
38 If ActiveCell.Column < 21 Then ActiveCell.Value = ActiveCell.Value & " " & Sheet2.
39 Range("SortAlphaAZ").Value Else: ActiveCell.Value = ActiveCell.Value & " " &
40 Sheet2.Range("SortNum19").Value 'Add in Descending Numerical or Alpha Character
41 End If 'End of Setting Sort Parameters
42 With .Sort 'With Sort on Reports sheet
43 .SetRange Sheet1.Range("Q9:X" & LastRow) 'Set Sort Range
44 .Apply 'Run Sort
45 End With 'End if With Sort
46 .Range("B1").ClearContents 'Clear Selected Row on Sort
47 .Range("B2").Value = False 'Set File Load To False
48 .Range("A1").Select 'Select A1 in order to unselect Range
49 End With 'End With Reports Sheet
50 ResetCalc 'Run Macro to Restart Calculation & Screen Updating
51 End Sub

```

A

ActCol, [30](#)
ActHeader, [30](#)
ActiveCell, [30](#)
Add, [30](#)
Apply, [30](#)

C

Cells, [30](#)
Clear, [30](#)
ClearContents, [30](#)
Column, [30](#)

D

DataOption, [30](#)

E

Explicit, [30](#)

F

FileSort, [30](#)

H

HeadCol, [30](#)

I

InStr, [30](#)

K

Key, [30](#)

L

LastRow, [30](#)

O

Order, [30](#)

R

Range, [30](#)
Replace, [30](#)
ResetCalc, [30](#)
Row, [30](#)

S

SetRange, [30](#)
Sheet1, [30](#)
Sheet2, [30](#)
Sort, [30](#)
SortFields, [30](#)
SortOn, [30](#)
SortType, [30](#)
StopCalc, [30](#)

V

Value, [30](#)

X

xlAscending, [30](#)
xlDescending, [30](#)
xlSortNormal, [30](#)
xlSortOnValues, [30](#)
xlUp, [30](#)

Thank You!

This source code was created and made available to help you gain a better understanding of how VBA is used to create amazing Excel-based applications.

Thank you so much for your continued shares, likes and support. It really helps.



Excel For Freelancers