

**Chapter 14 Sequential Files (Main Page)**

- 14.1 **DirListBox**, **FileListBox**, and **DriveListBox** toolbox icons.
- 14.2 Some **DirListBox**, **FileListBox**, and **DriveListBox** common properties and methods.
- 14.3 Some **FileListBox** properties.
- 14.4 Demonstrating controls **DirListBox**, **FileListBox**, **DriveListBox**.
- 14.5 Data hierarchy.
- 14.6 **Reference** dialog with **Microsoft Scripting Runtime** selected.
- 14.7 File System Objects (FSOs).
- 14.8 **FileSystemObject** methods.
- 14.9 Using a **FileSystemObject**.
- 14.10 Some common **File** properties and methods.
- 14.11 Demonstrating the **File** FSO.
- 14.12 Some **Folder** properties and methods.
- 14.13 Using a **Folder** FSO.
- 14.14 **Drive** properties.
- 14.15 Using a **Drive** FSO.
- 14.16 **TextStream** properties and methods.
- 14.17 Writing data to a text file.
- 14.18 File open modes.
- 14.19 Reading from a sequential file.
- 14.20 A credit inquiry program.



Fig. 14.1 **DirListBox**, **FileListBox** and **DriveListBox** toolbox icons.

Property/method	Description
<i>Properties</i>	
<b>DragIcon</b>	Icon which is displayed during drag-and-drop operation.
<b>DragMode</b>	<b>Integer</b> . Automatic or manual drag-and-drop.
<b>Enabled</b>	<b>Boolean</b> . Specifies whether or not the user is allowed to interact with the control.
<b>List</b>	<b>String</b> array. Array that stores the <b>Strings</b> that appear in the controls.
<b>ListCount</b>	<b>Integer</b> . Number of items in the <b>List</b> properties array.
<b>ListIndex</b>	<b>Integer</b> . Index of selected <b>List</b> property item. Index begins at 0 and is -1 when a value is not selected.
<b>MousePointer</b>	<b>Integer</b> . Specifies the shape of the mouse pointer when over the control.
<b>Visible</b>	<b>Boolean</b> . Specifies whether or not the control is visible.
<i>Methods</i>	
<b>Drag</b>	Starts, terminates, or aborts drag operations.
<b>Refresh</b>	Forces the control to repaint itself.
<b>SetFocus</b>	Transfers the focus to the control.

Fig. 14.2 Some **DirListBox**, **FileListBox** and **DriveListBox** common properties and methods.

Property	Description
<b>Archive</b>	<b>Boolean</b> . Specifies whether or not archive attributes are displayed (default is <b>True</b> ).
<b>Hidden</b>	<b>Boolean</b> . Specifies whether or not hidden attributes are displayed (default is <b>False</b> ).
<b>MultiSelect</b>	<b>Integer</b> . Specifies whether or not the user can make multiple selections (multiple selection is not allowed by default).
<b>Path</b>	<b>String</b> . Specifies the current path.
<b>Pattern</b>	<b>String</b> . Specifies the type of files displayed in the <b>FileListBox</b> .
<b>ReadOnly</b>	<b>Boolean</b> . Specifies whether or not read-only attributes are displayed.
<b>System</b>	<b>Boolean</b> . Specifies whether or not system attributes are displayed (default is <b>False</b> ).

Fig. 14.3 Some **FileListBox** properties.

---

```
1 ' Fig. 14.4
2 ' Demonstrating FileListBox, DirListBox,
3 ' and DriveListBox controls
4 Option Explicit      ' General declaration
5
6 Private Sub dirDirBox_Change()
7
8     ' Update the file path to the directory path
9     filFileBox.Path = dirDirBox.Path
10 End Sub
11
12 Private Sub drvDriveBox_Change()
13
14     On Error GoTo error handler
15
16     ' Update the directory path to the drive
17     dirDirBox.Path = drvDriveBox.Drive
18     Exit Sub
19
20 errorhandler:
21     Dim message As String
22
23     ' Check for device unavailable error
24     If Err.Number = 68 Then
25         Dim r As Integer
26
27         message = "Drive is not available."
28         r = MsgBox(message, vbRetryCancel + vbCritical, _
29             "VBHTP: Chapter 14")
30
31         ' Determine where control should resume
32         If r = vbRetry Then
33             Resume
34         Else ' Cancel was pressed.
35             drvDriveBox.Drive = drvDriveBox.List(1)
36             Resume Next
37         End If
38
39     Else
40         Call MsgBox(Err.Description, vbOKOnly + vbExclamation)
41         Resume Next
42     End If
43
44 End Sub
```

---

Fig. 14.4 Demonstrating controls **DirListBox**, **FileListBox** and **DriveListBox** (part 1 of 2)

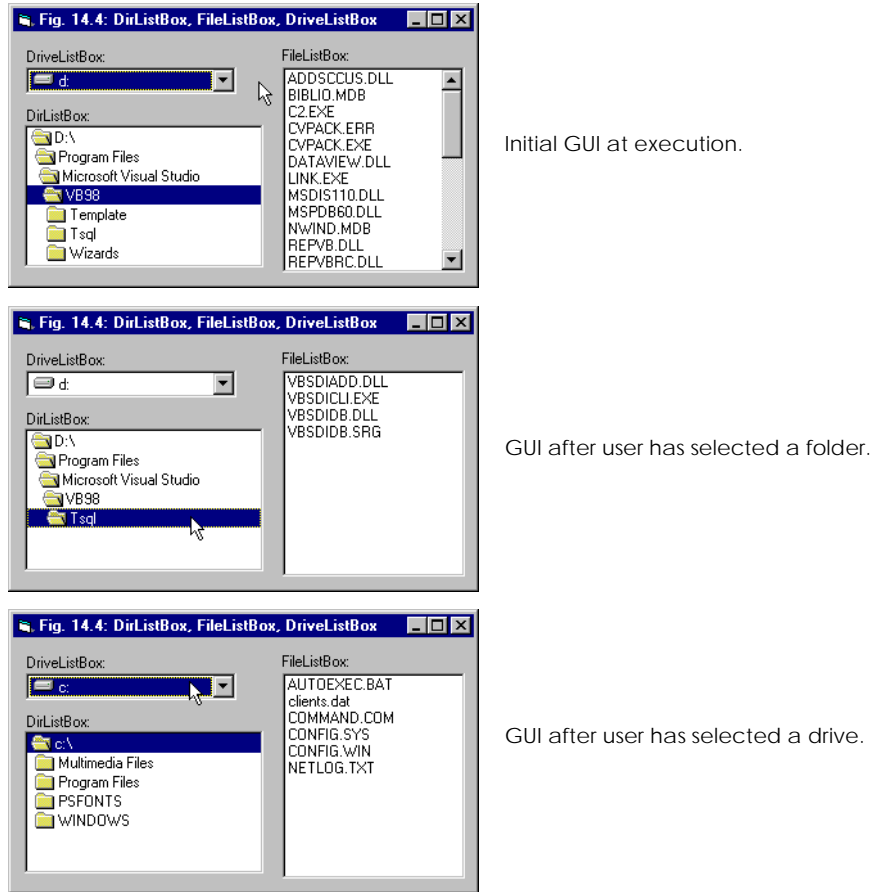


Fig. 14.4 Demonstrating controls **DirListBox**, **FileListBox** and **DriveListBox** (part 2 of 2).

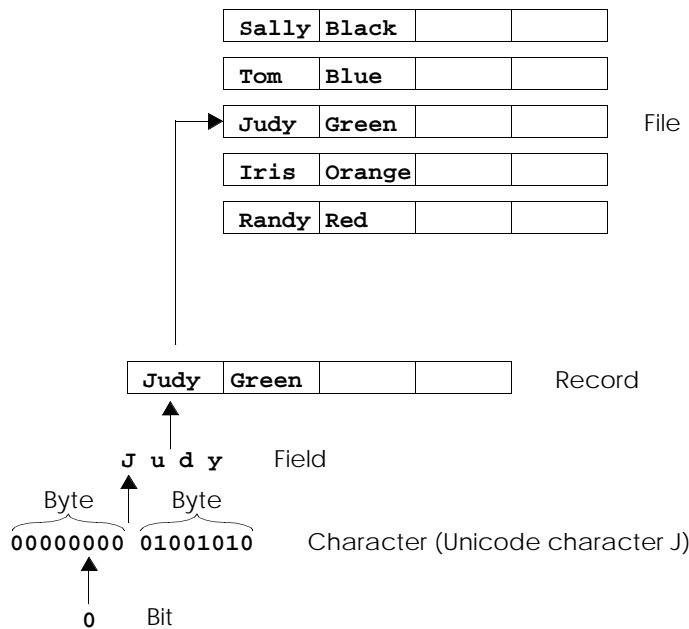


Fig. 14.5 Data hierarchy.

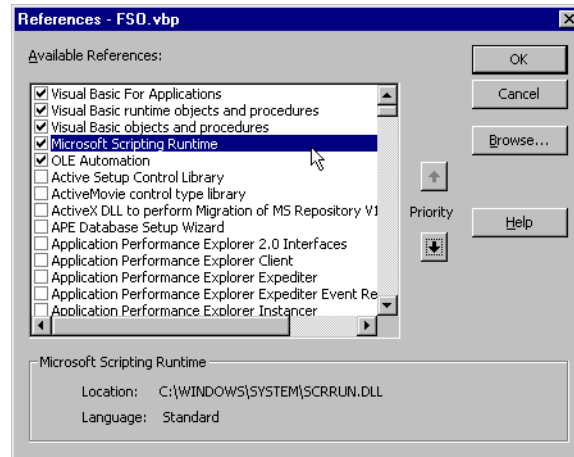


Fig. 14.6 References dialog with **Microsoft Scripting Runtime** selected.

Object type	Description
<b>FileSystemObject</b>	Allows the programmer to interact with <b>Files</b> , <b>Folders</b> and <b>Drives</b> .
<b>File</b>	Allows the programmer to manipulate <b>Files</b> .
<b>Folder</b>	Allows the programmer to manipulate <b>Folders</b> (i.e., directories).
<b>Drive</b>	Allows the programmer to gather information about <b>Drives</b> (hard disks, RAM disks—computer memory used as a substitute for hard disks to allow high speed file operations, CD-ROMs, etc.). <b>Drives</b> can be local or remote.
<b>TextStream</b>	Allows the programmer to read and write text files.

Fig. 14.7 File System Objects (FSOs).

Methods	Description
<b>CopyFile</b>	Copies an existing <b>File</b> .
<b>CopyFolder</b>	Copies an existing <b>Folder</b> .
<b>CreateFolder</b>	Creates and returns a <b>Folder</b> .
<b>CreateTextFile</b>	Creates and returns a text <b>File</b> .
<b>DeleteFile</b>	Deletes a <b>File</b> .
<b>DeleteFolder</b>	Deletes a <b>Folder</b> .
<b>DriveExists</b>	Tests whether or not a <b>Drive</b> exists.
<b>FileExists</b>	Tests whether or not a <b>File</b> exists. Returns <b>Boolean</b> .
<b>FolderExists</b>	Tests whether or not a <b>Folder</b> exists. Returns <b>Boolean</b> .
<b>GetAbsolutePathName</b>	Returns the absolute path as a <b>String</b> .
<b>GetDrive</b>	Returns the specified <b>Drive</b> .

Fig. 14.8 **FileSystemObject** methods (part 1 of 2).

Methods	Description
<code>GetDriveName</code>	Returns the <b>Drive</b> drive name.
<code>GetFile</code>	Returns the specified <b>File</b> .
<code>GetFileName</code>	Returns the <b>File</b> file name.
<code>GetFolder</code>	Returns the specified <b>Folder</b> .
<code>GetParentFolderName</code>	Returns a <b>String</b> representing the parent folder name.
<code>GetTempName</code>	Creates and returns a <b>String</b> representing a file name.
<code>MoveFile</code>	Moves a <b>File</b> .
<code>MoveFolder</code>	Moves a <b>Folder</b> .
<code>OpenTextFile</code>	Opens an existing text <b>File</b> . Returns a <b>TextStream</b> .

Fig. 14.8 `FileSystemObject` methods (part 2 of 2).

```

1  ' Fig. 14.9
2  ' Demonstrating FileSystemObjects
3  Option Explicit          ' General declaration
4  Dim mFileSysObj As New FileSystemObject ' General declaration
5
6  Private Sub dirDirBox_Change()
7
8      ' Update the file path to the directory path
9      filFileBox.Path = dirDirBox.Path
10 End Sub
11
12 Private Sub drvDriveBox_Change()
13
14     On Error GoTo error handler
15
16     ' Update the directory path to the drive
17     dirDirBox.Path = drvDriveBox.Drive
18     Exit Sub
19
20 errorhandler:
21     Dim message As String
22
23     ' Check for device unavailable error
24     If Err.Number = 68 Then
25         Dim r As Integer
26
27         message = "Drive is not available."

```

Fig. 14.9 Using a `FileSystemObject` (part 1 of 4).

```

28     r = MsgBox(message, vbRetryCancel + vbCritical, _
29         "VBHTP: Chapter 14")
30
31     ' Determine where control should resume
32     If r = vbRetry Then
33         Resume
34     Else ' Cancel was pressed.
35         drvDriveBox.Drive = drvDriveBox.List(1)
36         Resume Next
37     End If
38
39 Else
40     Call MsgBox(Err.Description, vbOKOnly + vbExclamation)

```

```

41         Resume Next
42     End If
43
44 End Sub
45
46 Private Sub filFileBox_Click()
47     Call displayData      ' Update TextBox
48 End Sub
49
50 ' Programmer defined
51 Private Sub displayData()
52     txtDisplay.Text = ""
53     txtDisplay.Text = "GetAbsolutePathName: " & _
54         mFileSysObj.GetAbsolutePathName( _
55         filFileBox.Path) & vbNewLine & _
56         "GetDriveName: " & _
57         mFileSysObj.GetDriveName( _
58         filFileBox.Path) & vbNewLine & _
59         "GetParentFolderName: " & _
60         mFileSysObj.GetParentFolderName( _
61         filFileBox.Path) & vbNewLine & _
62         "GetTempName: " & mFileSysObj.GetTempName
63 End Sub
64
65 Private Sub mnuitmCreateFolder_Click()
66     Dim s As String
67
68     ' Get the Folder name
69     s = InputBox("Enter complete path and folder name:", "CREATE")
70
71     ' Test if the Folder already exists
72     If mFileSysObj.FolderExists(s) Then
73         Call MsgBox("Folder already exists! Cannot create.")
74         Exit Sub
75     End If
76
77     Call mFileSysObj.CreateFolder(s)      ' Create the Folder
78     Call dirDirBox.Refresh                ' Repaint DirListBox
79 End Sub
80

```

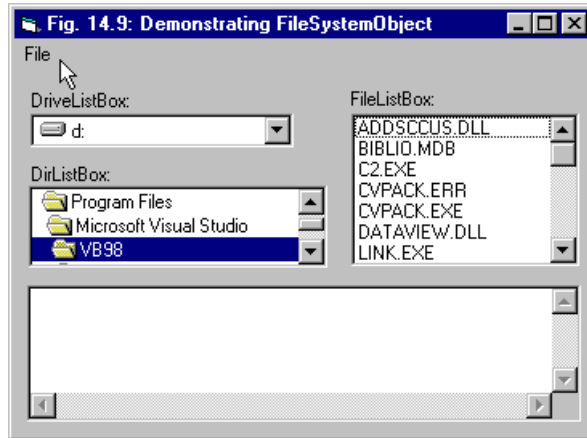
---

Fig. 14.9 Using a **FileSystemObject** (part 2 of 4).

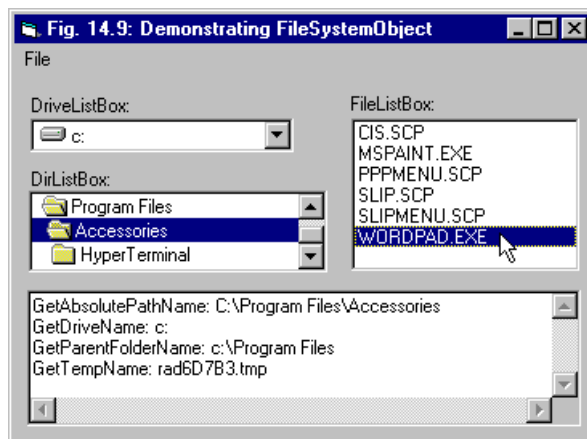
```

81 Private Sub mnuitmDeleteFolder_Click()
82     Dim s As String
83
84     ' Get the Folder name
85     s = InputBox("Enter complete path and folder name:", "DELETE")
86
87     ' Test if the Folder already exists
88     If mFileSysObj.FolderExists(s) = False Then
89         Call MsgBox("Folder does not exist! Cannot delete.")
90         Exit Sub
91     End If
92
93     Call mFileSysObj.DeleteFolder(s)      ' Delete the Folder
94     Call dirDirBox.Refresh                ' Repaint DirListBox
95 End Sub
96
97 Private Sub mnuitmExit_Click()
98     End      ' Terminate execution
99 End Sub

```

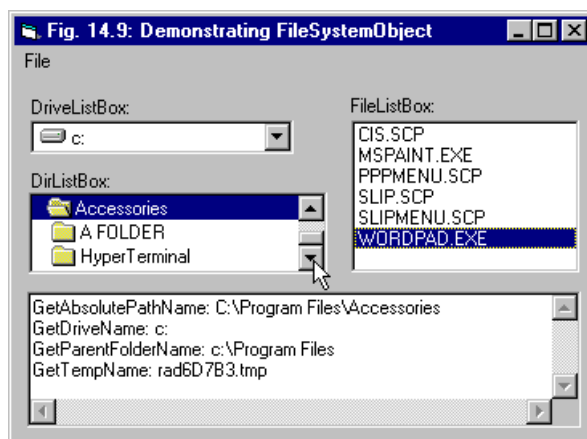


Initial GUI at execution.



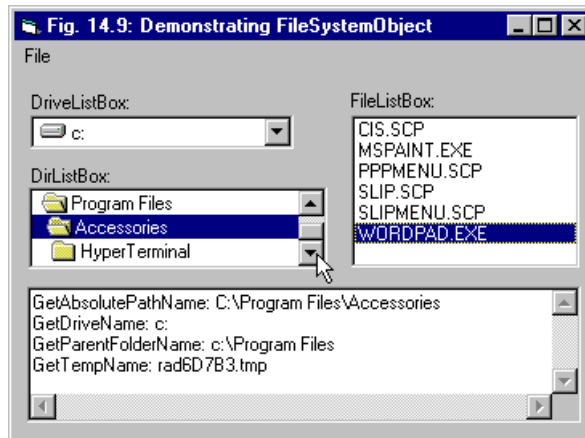
GUI after user has selected **c:** drive and the **Accessories** menu. The **TextBox** displays information about file **WORDPAD.EXE** which the user clicked.

Fig. 14.9 Using a **FileSystemObject** (part 3 of 4).



GUI after the user selects **Create Folder** from the **File** menu and enters **c:\program files\accessories\A FOLDER**. Note that the folder is automatically inserted in alphabetical order.





GUI after the user selects **Delete Folder** from the **File** menu and enters **c:\program files\accessories\A FOLDER**.

Fig. 14.9 Using a **FileSystemObject** (part 4 of 4).

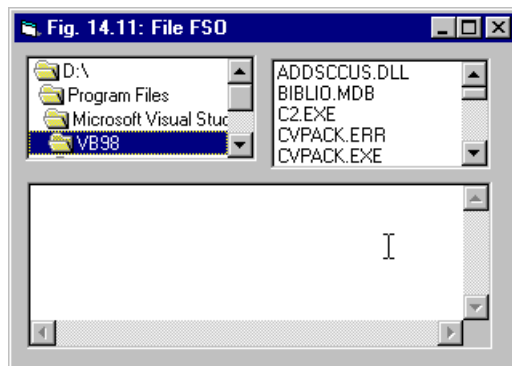
Property/method	Description
<i>Properties</i>	
<b>DateCreated</b>	<b>Date.</b> The date the <b>File</b> was created.
<b>DateLastAccessed</b>	<b>Date.</b> The date the <b>File</b> was last accessed.
<b>DateLastModified</b>	<b>Date.</b> The date the <b>File</b> was last modified.
<b>Drive</b>	<b>Drive.</b> The <b>Drive</b> where the file is located.
<b>Name</b>	<b>String.</b> The <b>File</b> name.
<b>ParentFolder</b>	<b>String.</b> The <b>File</b> 's parent folder name.
<b>Path</b>	<b>String.</b> The <b>File</b> 's path.
<b>ShortName</b>	<b>String.</b> The <b>File</b> 's name expressed as a short name.
<b>Size</b>	The size of the <b>File</b> in bytes.
<i>Methods</i>	
<b>Copy</b>	Copy this <b>File</b> . Same as <b>FileSystemObject</b> method <b>CopyFile</b> .
<b>Delete</b>	Delete this <b>File</b> . Same as <b>FileSystemObject</b> method <b>DeleteFile</b> .
<b>Move</b>	Move this <b>File</b> . Same as <b>FileSystemObject</b> method <b>MoveFile</b> .
<b>OpenAsTextStream</b>	Opens an existing <b>File</b> as a text <b>File</b> . Returns <b>TextStream</b> .

Fig. 14.10 Some common **File** properties and methods.

```

1  ' Fig. 14.11
2  ' Demonstrating File FSO
3  Option Explicit          ' General declaration
4  Dim mFso As New FileSystemObject ' General declaration
5
6  Private Sub dirDirectory_Change()
7
8      ' Update filFile's Path when dirDirectory changes
9      filFile.Path = dirDirectory.Path
10 End Sub
11
12 Private Sub filFile_Click()
13     Dim theFile As File
14
15     txtBox.Text = ""      ' Clear TextBox
16
17     ' Determine which file name was selected and return
18     ' its File object.
19     Set theFile = mFso.GetFile(filFile.Path & "\" & _
20         filFile.List(filFile.ListIndex))
21
22     ' Display File information in PictureBox
23     txtBox.Text = "Created: " & theFile.DateCreated & _
24         vbNewLine & _
25         "Last Accessed: " & theFile.DateLastAccessed & _
26         vbNewLine & "Last Modified: " & _
27         theFile.DateLastModified & vbNewLine & _
28         "Drive: " & theFile.Drive & vbNewLine & _
29         "Size: " & theFile.Size & " bytes" & _
30         vbNewLine & _
31         "Path: " & theFile.Path & vbNewLine & _
32         "Short Name: " & theFile.ShortName
33 End Sub

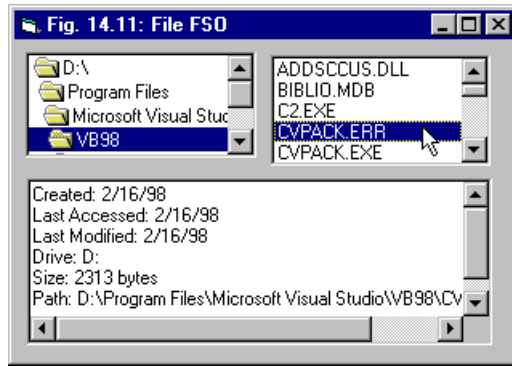
```



Initial GUI at execution.

---

**Fig. 14.11** Demonstrating the **File** FSO (part 1 of 2).



GUI after programmer clicks a file.

Fig. 14.11 Demonstrating the **File** FSO (part 2 of 2).

Property/method	Description
<i>Properties</i>	
<b>Attributes</b>	<b>Integer</b> . Value corresponding to <b>Folder</b> 's attributes (read only, hidden, etc.)
<b>DateCreated</b>	<b>Date</b> . The date the file was created.
<b>DateLastAccessed</b>	<b>Date</b> . The date the file was last accessed.
<b>DateLastModified</b>	<b>Date</b> . The date the file was last modified.
<b>Drive</b>	<b>Drive</b> . The <b>Drive</b> where the folder is located.
<b>IsRootFolder</b>	<b>Boolean</b> . Indicates whether or not a <b>Folder</b> is the root folder.
<b>Name</b>	<b>String</b> . The <b>Folder</b> 's name.
<b>ParentFolder</b>	<b>String</b> . The <b>Folder</b> 's parent folder name.
<b>Path</b>	<b>String</b> . The <b>Folder</b> 's path.
<b>ShortName</b>	<b>String</b> . The <b>Folder</b> 's name expressed as a short name.
<b>ShortPath</b>	<b>String</b> . The <b>Folder</b> 's path expressed as a short path.
<b>Size</b>	<b>VARIANT</b> . The total size in bytes of all subfolders and files.
<b>Type</b>	<b>String</b> . The <b>Folder</b> type.
<i>Methods</i>	
<b>Delete</b>	Delete this <b>Folder</b> . Same as <b>FileSystemObject</b> method <b>DeleteFile</b> .
<b>Move</b>	Move this <b>Folder</b> . Same as <b>FileSystemObject</b> method <b>MoveFolder</b> .
<b>Copy</b>	Copy this <b>Folder</b> . Same as <b>FileSystemObject</b> method <b>CopyFolder</b> .

Fig. 14.12 Some **Folder** properties and methods.

```

1  ' Fig. 14.13
2  ' Demonstrating the Folder FSO
3  Option Explicit          ' General declaration
4  Dim mFso As New FileSystemObject ' General declaration
5
6  Private Sub dirBox_Click()
7      Dim f As Folder
8

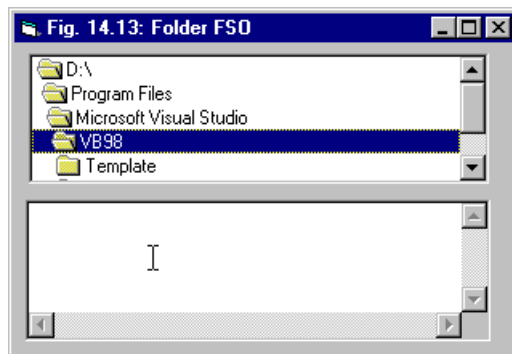
```

Fig. 14.13 Using a **Folder** FSO (part 1 of 2).

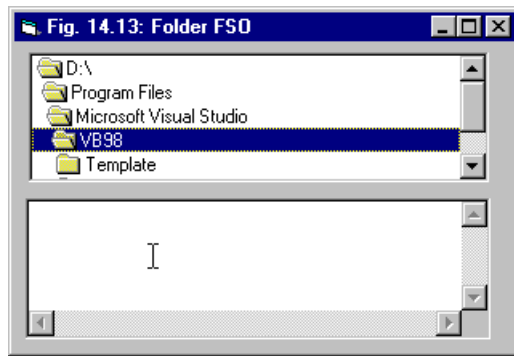
```

9      ' Get a Folder to the selected item
10     Set f = mFso.GetFolder(dirBox.List(dirBox.ListIndex))
11
12     txtDisplay.Text = ""          ' Clear TextBox
13
14     ' Test for the root folder
15     If f.IsRootFolder = False Then
16         txtDisplay.Text = "Root folder: " & f.IsRootFolder _
17             & vbNewLine & _
18             "Parent Folder: " & f.ParentFolder & _
19             vbNewLine & "Size: " & f.Size
20     Else
21         txtDisplay.Text = "Root folder: " & f.IsRootFolder
22     End If
23
24     txtDisplay.Text = txtDisplay.Text & vbNewLine & "Type: " & _
25         f.Type & vbNewLine & _
26         "Short Path: " & f.ShortPath & _
27         vbNewLine & "Path: " & f.Path & _
28         vbNewLine & "Short Name: " & f.ShortName
29 End Sub

```



Initial GUI at execution.



Initial GUI at execution.

---

Fig. 14.13 Using a **Folder** FSO (part 2 of 2).

Property	Description
<b>AvailableSpace</b>	<b>Variant.</b> The amount of available <b>Drive</b> space in bytes.
<b>DriveLetter</b>	<b>String.</b> The letter assigned the <b>Drive</b> (e.g., "C").
<b>DriveType</b>	<b>Integer.</b> The <b>Drive</b> type. Constants <b>Unknown</b> , <b>Removable</b> , <b>Fixed</b> , <b>Remote</b> , <b>CDRom</b> and <b>RamDisk</b> represent <b>Drive</b> types and have the values 0 - 5, respectively.
<b>FileSystem</b>	<b>String.</b> The file system <b>Drive</b> description (FAT, FAT32, NTFS, etc.).
<b>FreeSpace</b>	<b>Variant.</b> Same as <b>AvailableSpace</b> .
<b>IsReady</b>	<b>Boolean.</b> Indicates whether or not a <b>Drive</b> is ready for use.
<b>Path</b>	<b>String.</b> The <b>Drive's Path</b> .
<b>RootFolder</b>	<b>Folder.</b> The <b>Drive's root Folder</b> .
<b>SerialNumber</b>	<b>Long.</b> The <b>Drive</b> serial number.
<b>TotalSize</b>	<b>Variant.</b> The total <b>Drive</b> size in bytes.
<b>VolumeName</b>	<b>String.</b> The <b>Drive</b> volume name.

Fig. 14.14 Drive properties.

```

1  ' Fig. 14.15
2  ' Demonstrating the Drive FSO
3  Option Explicit          ' General declaration
4  Dim mFso As New FileSystemObject ' General Declaration
5
6  Private Sub Form_Load()
7      Call drvBox_Change    ' Call DriveListBox Change
8  End Sub
9
10 Private Sub drvBox_Change()
11     Dim d As Drive
12
13     ' Get the Drive and assign it to d
14     Set d = mFso.GetDrive(drvBox.List(drvBox.ListIndex))
15
16     ' Enable error handler
17     On Error GoTo errhandler
18     lblStats(0).Caption = "Letter: " & d.DriveLetter & _
19                          Space$(3) & "S/N #: " & _
20                          & d.SerialNumber & Space$(3) & _
21                          "Free space: " & d.FreeSpace / _
22                          1000000 & " MB"
23     lblStats(1).Caption = "Type: " & d.DriveType & _
24                          Space$(3) & "File System: " & _
25                          & d.FileSystem & Space$(3) & _
26                          "Total Size: " & d.TotalSize / _
27                          1000000 & " MB"
28     Exit Sub
29
30 errhandler:
31     Call MsgBox(Err.Description, vbCritical, "ERROR")
32     drvBox.Drive = drvBox.List(1)

```

```

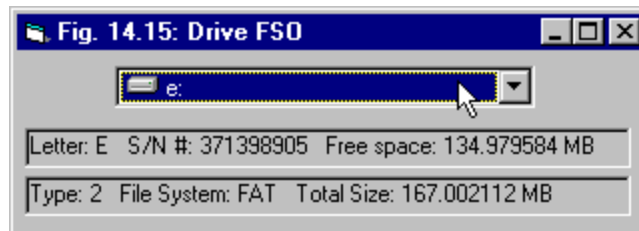
33     Exit Sub
34 End Sub

```



Initial GUI at execution.

Fig. 14.15 Using a **Drive** FSO (part 1 of 2).



GUI after user selects **e:** drive.



GUI after user selects **a:** drive.

Fig. 14.15 Using a **Drive** FSO (part 2 of 2).

Property/method	Description
<i>Properties</i>	
<b>AtEndOfLine</b>	<b>Boolean.</b> Specifies if the position is at the end of the line.
<b>AtEndOfStream</b>	<b>Boolean.</b> Specifies if the position is at the end of the stream.
<b>Column</b>	<b>Long.</b> The current character's position in.
<b>Line</b>	<b>Long.</b> The current line number.
<i>Methods</i>	
<b>Close</b>	Closes the <b>TextStream</b> .
<b>Read</b>	Reads a specified number of characters from the <b>TextStream</b> . Returns a <b>String</b> .
<b>ReadAll</b>	Reads the entire <b>TextStream</b> into a <b>String</b> . Returns a <b>String</b> .
<b>ReadLine</b>	Reads a line from a <b>TextStream</b> . Returns a <b>String</b> .
<b>Skip</b>	Skips a specified number of <b>TextStream</b> characters.
<b>SkipLine</b>	Skips a line.

Fig. 14.16 **TextStream** properties and methods (part 1 of 2).

Property/method	Description
Write	Writes a <b>String</b> to a <b>TextStream</b> .
WriteBlankLines	Writes a specified number of blank lines to a <b>TextStream</b> .
WriteLine	Writes a <b>String</b> followed by an end-of-line character to a <b>TextStream</b> .

Fig. 14.16 **TextStream** properties and methods (part 2 of 2).

```

1  ' Fig. 14.17
2  ' Writing to a sequential text file
3  Option Explicit                ' General declaration
4  Dim mFileSysObj As New FileSystemObject ' General declaration
5  Dim mFile As File              ' General declaration
6  Dim mTxtStream As TextStream  ' General declaration
7
8  Private Sub Form_Load()
9
10     ' Create a text file
11     Call mFileSysObj.CreateTextFile("c:\clients.dat")
12
13     ' Once file is created, reference the file
14     Set mFile = mFileSysObj.GetFile("c:\clients.dat")
15
16     ' Open a text stream for writing to the file
17     Set mTxtStream = mFile.OpenAsTextStream(ForWriting)
18
19     ' Display path in lblFileName
20     lblFileName.Caption = mFile.Path
21 End Sub

```

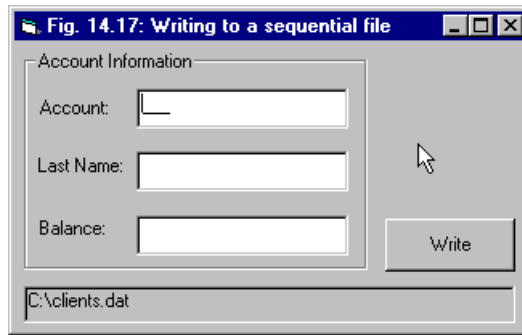
Fig. 14.17 Writing data to a text file (part 1 of 3).

```

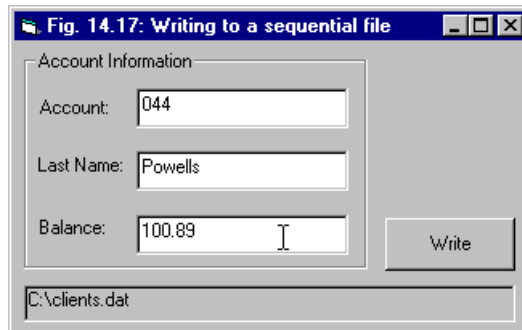
22
23 Private Sub cmdWrite_Click()
24
25     ' Write the data to the file
26     Call mTxtStream.WriteLine(mskAccount.Text & " " & _
27                               txtName.Text & " " & _
28                               txtBalance.Text)
29
30     ' Clear MaskEdit and TextBoxes
31     txtName.Text = ""
32     txtBalance.Text = ""
33
34     ' Set several properties for mskAccount
35     ' using With statement
36     With mskAccount
37         .Text = "000"        ' Display all zeros in MaskEdit
38         .SelStart = 0       ' Start highlighting at position 0
39         .SelLength = 3      ' Highlight 3 characters
40         .SetFocus           ' Transfer focus
41     End With
42 End Sub
43
44 Private Sub Form_Terminate()
45     Call mTxtStream.Close    ' Close the text stream
46 End Sub

```



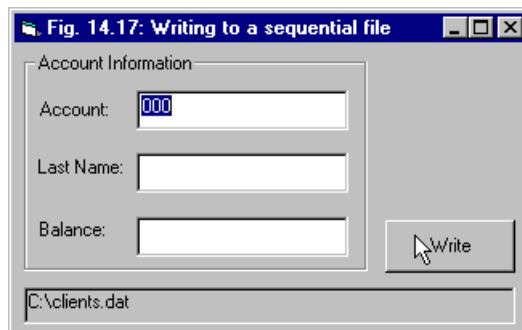


Initial GUI at execution.



GUI as programmer is entering account information.

Fig. 14.17 Writing data to a text file (part 2 of 3).



GUI after user clicks **Write**. Focus shifts to **MaskEdit** where the three digits are selected.

Fig. 14.17 Writing data to a text file (part 3 of 3).

File Open Mode	Description
<b>ForReading</b>	Open the file for reading only. Data cannot be written to the file.
<b>ForWriting</b>	Open a file for writing. If the file already contains data, the data are truncated—all data in the file are discarded.
<b>ForAppending</b>	Open a file to allow writing at the end of the file.

Fig. 14.18 File open modes.

```

1  ' Fig. 14.19
2  ' Reading from a sequential text file
3  Option Explicit          ' General declaration
4  Dim mFileSysObj As New FileSystemObject ' General declaration
5  Dim mFile As File       ' General declaration
6  Dim mTxtStream As TextStream ' General declaration
7
8  Private Sub Form_Load()
9
10     ' Get the file
11     Set mFile = mFileSysObj.GetFile("c:\clients.dat")
12
13     ' Open a text stream for writing to the file
14     Set mTxtStream = mFile.OpenAsTextStream(ForReading)
15
16     ' Display path in lblFileName
17     lblFileName.Caption = mFile.Path
18 End Sub
19
20 Private Sub cmdRead_Click()
21     Dim s As String
22
23     On Error GoTo handler ' Set error trap
24
25     ' Read the data
26     s = mTxtStream.ReadLine
27
28     ' Parse String s to get values for MaskEdit
29     ' and TextBoxes
30     Dim mark1 As Integer, mark2 As Integer
31
32     mark1 = 4 ' Location of first space in s
33
34     ' Place only the String portion representing the
35     ' account in the MaskEdit.
36     mskAccount.Text = Trim$(Mid$(s, 1, mark1))
37
38     ' Position to first letter of name
39     mark1 = mark1 + 1
40
41     ' Determine location of second space character and
42     ' add 1 to include the space.
43     mark2 = InStr(mark1, s, " ", vbTextCompare) + 1
44
45     ' Place only the String portion representing the
46     ' name in the TextBox.
47     txtName.Text = Trim$(Mid$(s, mark1, mark2 - mark1))
48
49     ' Place the formatted dollar amount in the TextBox.
50     ' mark2 is positioned at the beginning of the amount.
51     txtBalance.Text = Format$(Mid$(s, mark2, Len(s) - mark1), _
52         "Currency")
53     Exit Sub

```

Fig. 14.19 Reading from a sequential file (part 1 of 2).

```

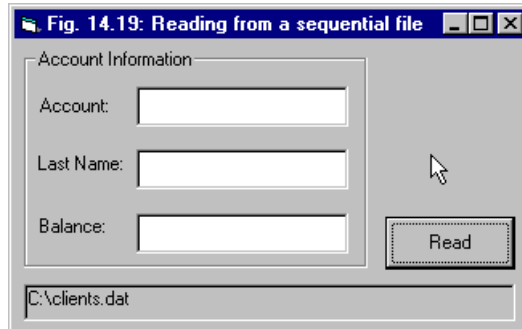
54
55 handler:
56     If Err.Number = 62 Then ' EOF error
57         Call mTxtStream.Close
58         cmdRead.Enabled = False
59         lblFileName.Caption = ""

```

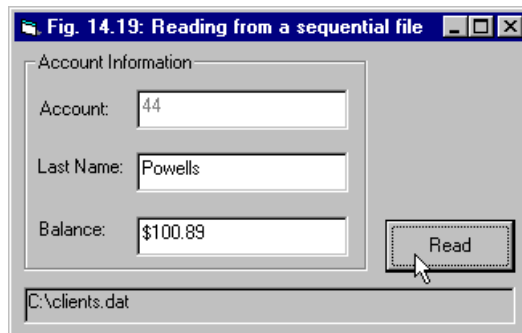
```

60     Else
61         Call MsgBox(Err.Description)
62     End If
63
64 End Sub

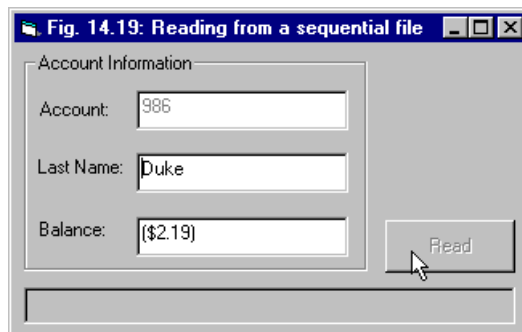
```



Initial GUI at execution.



GUI after user clicks the **Read** button once.



GUI when the last set of information is read.

Fig. 14.19 Reading from a sequential file (part 2 of 2).

---

```

1  ' Fig. 14.20
2  ' Credit inquiry program
3  Option Explicit                ' General declaration
4  Dim mFso As New FileSystemObject ' General declaration
5  Dim mType As Integer           ' General declaration
6  Const mCREDIT = 0, mDEBIT = 1, mZERO = 2 ' General declaration
7
8  Private Sub cmdButton_Click(Index As Integer)
9      mType = Index ' Assign cmdButton control array index
10     Call openAndReadFile ' Open and read file
11 End Sub
12
13 Private Sub openAndReadFile()
14     Dim txtStream As TextStream, s As String
15     Dim balance As Currency, pos As Long
16
17     txtDisplay.Text = "Accounts:"
18
19     ' Get "clients.dat" and open a TextStream for reading
20     Set txtStream = mFso.GetFile("c:\clients.dat"). _
21         OpenAsTextStream(ForReading)
22

```

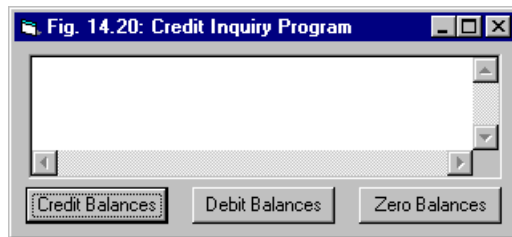
---

Fig. 14.20 A credit inquiry program (part 1 of 3).

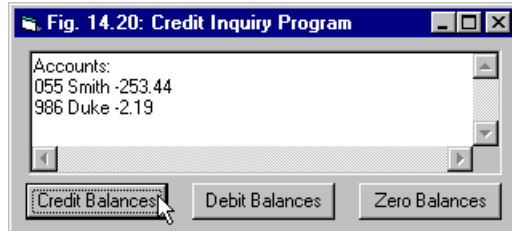
```

23     ' Loop until end of stream is found
24     Do
25         s = txtStream.ReadLine ' Read one line
26
27         ' Find the position of the second space
28         pos = InStr(InStr(1, s, " ", vbTextCompare) + 1, s, " ", _
29             vbTextCompare)
30
31         ' Extract the String that contains the balance
32         balance = Trim$(Mid$(s, pos, Len(s) - pos))
33
34         ' Determine what if anything should be displayed
35         If (mType = mCREDIT And balance < 0) Then
36             txtDisplay.Text = txtDisplay.Text & vbCrLf & s
37         ElseIf (mType = mDEBIT And balance > 0) Then
38             txtDisplay.Text = txtDisplay.Text & vbCrLf & s
39         ElseIf (mType = mZERO And balance = 0) Then
40             txtDisplay.Text = txtDisplay.Text & vbCrLf & s
41         End If
42
43     Loop While (txtStream.AtEndOfStream = False)
44
45     Call txtStream.Close ' Close TextStream
46 End Sub

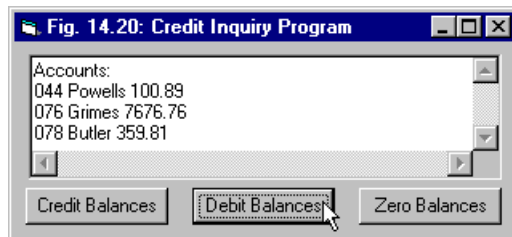
```



Initial GUI at execution.



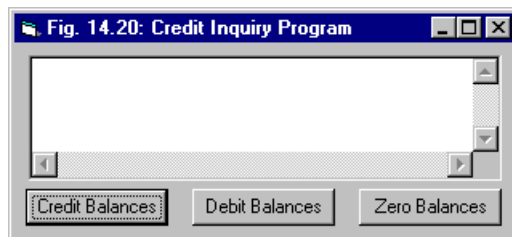
GUI after user has pressed the **Credit Balances** button.



GUI after user has pressed the **Debit Balances** button.

---

Fig. 14.20 A credit inquiry program (part 2 of 3).



Initial GUI at execution.

---

Fig. 14.20 A credit inquiry program (part 3 of 3).