



[VB6 beginners tutorial](#) -
Learn VB6

[Advanced VB6 tutorial](#) -
Learn Advanced VB6

[Systems Analysis](#) -
System analysis and
Design tutorial for
Software Engineering

Browse Topics

- [Getting started](#)
- [Data Types](#)
- [Modules](#)
- [Operators in VB6](#)
- [VB6 Variable](#)
- [VB6 Procedures](#)
- [VB6 Control Structures](#)
- [Loops in VB6](#)
- [VB6 Exit Do & With End With](#)
- [Arrays in VB6](#)
- [User-Defined Data Types](#)
- [VB6 Constants](#)
- [VB6 Built-in Functions](#)
- [Date and Time in VB6](#)
- [VB6 Controls](#)
- [TextBox Control](#)
- [ComboBox & OptionButton](#)
- [Label & Frame](#)
- [PictureBox & ImageBox](#)
- [Timer Control](#)
- [ListBox & ComboBox](#)
- [VB6 ScrollBar](#)
- [Control Arrays in VB6](#)
- [Files controls in VB6](#)
- [VB6 CheckBox](#)
- [Forms in VB6](#)
- [Menus in VB6](#)
- [MDI Form in VB6](#)
- [InputBox](#)
- [MessageBox](#)
- [Mouse events](#)
- [Mouse Move](#)
- [Error Handling](#)
- [Error Handling \(2\)](#)
- [VB6 Database](#)

You are here: [Visual Basic](#) > **[VB6 \(Beginners Tutorial\)](#)**

[Previous Page](#) | [Table of Contents](#) | [Next Page](#)

DriveListBox, DirListBox, And FileListBox Controls In Visual Basic 6

Three of the controls on the ToolBox let you access the computer's file system. They are DriveListBox, DirListBox and FileListBox controls (see below figure), which are the basic blocks for building dialog boxes that display the host computer's file system. Using these controls, user can traverse the host computer's file system, locate any folder or files on any hard disk, even on network drives. The files are controls are independent of one another, and each can exist on it's own, but they are rarely used separately. The files controls are described next.

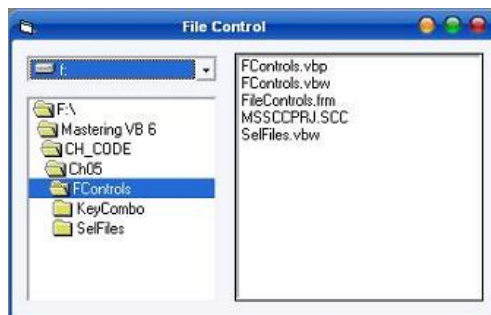
In a nutshell, the DriveListBox control is a combobox-like control that's automatically filled with your drive's letters and volume labels. The DirListBox is a special list box that displays a directory tree. The FileListBox control is a special-purpose ListBox control that displays all the files in a given directory, optionally filtering them based on their names, extensions, and attributes.

These controls often work together on the same form; when the user selects a drive in a DriveListBox, the DirListBox control is updated to show the directory tree on that drive. When the user selects a path in the DirListBox control, the FileListBox control is filled with the list of files in that directory. These actions don't happen automatically, however—you must write code to get the job done.

After you place a DriveListBox and a DirListBox control on a form's surface, you usually don't have to set any of their properties; in fact, these controls don't expose any special property, not in the Properties window at least. The FileListBox control, on the other hand, exposes one property that you can set at design time—the Pattern property. This property indicates which files are to be shown in the list area: Its default value is *.* (all files), but you can enter whatever specification you need, and you can also enter multiple specifications using the semicolon as a separator. You can also set this property at run time, as in the following line of code:

```
File1.Pattern = "*.txt;*.doc;*.rtf"
```

Following figure shows three file controls are used in the design of Forms that let users explore the entire structure of their hard disks.



- **DriveListBox** : Displays the names of the drives within and connected to the PC. The basic property of this control is the drive property, which set the drive to be initially selected in the control or returns the user's selection.
- **DirListBox** : Displays the folders of current Drive. The basic property of this control is the Path property, which is the name of the folder whose sub folders are displayed in the control.
- **FileListBox** : Displays the files of the current folder. The basic property of this control is also called Path, and it's the path name of the folder whose files are displayed.

The three File controls are not tied to one another. If you place all three of them on a Form, you will see the names of all the folders under the current folder, and so on. Each time you select a folder in the DirListBox by double clicking its name, its sub folders are displayed. Similarly, the FileListBox control will display the names of all files in the current folder. Selecting a drive in the DriveListBox control, however this doesn't affect the contents of the DirListBox.

To connect to the File controls, you must assign the appropriate values to the properties. To compel the DirListBox to display the folders of the selected drive in the DriveListBox, you must make sure that each time the user selects another drive, the Path property of the DirListBox control matches the Drive property of the DriveListBox.

After these preliminary steps, you're ready to set in motion the chain of events. When the user selects a new drive in the DriveListBox control, it fires a Change event and returns the drive letter (and volume label) in its Drive property. You trap this event and set the DirListBox control's Path property to point to the root directory of the selected drive:

```
Private Sub Drive1_Change()  
' The Drive property also returns the volume label, so trim it.  
Dir1.Path = Left$(Drive1.Drive, 1) & ":\"  
End Sub
```

When the user double-clicks on a directory name, the DirListBox control raises a Change event; you trap this event to set the FileListBox's Path property accordingly:

```
Private Sub Dir1_Change()  
File1.Path = Dir1.Path  
End Sub
```

Finally, when the user clicks on a file in the FileListBox control, a Click event is fired (as if it were a regular ListBox control), and you can query its Filename property to learn which file has been selected. Note how you build the complete path:

```
Filename = File1.Path  
If Right$(Filename, 1) <> "\" Then Filename = Filename & "\"  
Filename = Filename & File1.Filename
```

This website uses cookies to ensure you get the best experience on our website. [learn more](#) [Got It!](#)

including the ListCount and the ListIndex properties and the Scroll event. The FileListBox control supports multiple selection, hence you can set its MultiSelect property in the Properties window and query the SelCount and Selected properties at run time.

The FileListBox control also exposes a few custom Boolean properties, Normal, Archive, Hidden, ReadOnly, and System, which permit you to decide whether files with these attributes should be listed. (By default, the control doesn't display hidden and system files.) This control also supports a couple of custom events, PathChange and PatternChange, that fire when the corresponding property is changed through code. In most cases, you don't have to worry about them, and I won't provide examples of their usage.

The problem with the DriveListBox, DirListBox and FileListBox controls is that they're somewhat outdated and aren't used by most commercial applications any longer. Moreover, these controls are known to work incorrectly when listing files on network servers and sometimes even on local disk drives, especially when long file and directory names are used. For this reason, I discourage you from using them and suggest instead that you use the Common Dialog controls for your FileOpen and FileSave dialog boxes. But if you need to ask the user for the name of a directory rather than a file, you're out of luck because—while Windows does include such a system dialog box, named BrowseForFolders dialog—Visual Basic still doesn't offer a way to display it (unless you do some advanced API programming). Fortunately, Visual Basic 6 comes with a new control—the ImageCombo control—that lets you simulate the appearance of the DriveListBox control. It also offers you a powerful library—the FileSystemObject library—that completely frees you from using these three controls, if only as hidden controls that you use just for quickly retrieving information on the file system.

See Also

- [TextBox Control in Visual Basic 6](#)
- [CommandButton and OptionButton Controls in Visual Basic 6](#)
- [CheckBox control in Visual Basic 6](#)
- [Label and Frame Controls in Visual Basic 6](#)
- [PictureBox and Image Controls in Visual Basic 6](#)
- [The Timer, Line, Shape and OLE Controls in Visual Basic 6](#)
- [Working with Menus in Visual Basic 6](#)
- [ScrollBar Control in Visual Basic 6](#)

[Previous Page](#) | [Table of Contents](#) | [Next Page](#)

[Home](#) | [About Us](#) | [Privacy Policy](#) | [Contact Us](#)

Copyright © [Freetutes.com](#) | All Rights Reserved