



Ordering Information:

[Visual Basic 6 How to Program](#)
[The Complete Visual Basic 6 Training Course](#)

- View the complete [Table of Contents](#)
- Read the [Preface](#)
- Download the [Code Examples](#)

To view all the Deitel products and services available, visit the Deitel Kiosk on InformIT at www.informIT.com/deitel.

To follow the Deitel publishing program, sign-up now for the *DEITEL™ BUZZ ON-LINE* e-mail newsletter at www.deitel.com/newsletter/subscribeinformIT.html.

To learn more about our [Visual Basic 6 and Visual Basic .NET training courses](#) or any other Deitel instructor-led corporate training courses that can be delivered at your location, visit www.deitel.com/training, contact our Director of Corporate Training Programs at (978) 461-5880 or e-mail: christi.kelsey@deitel.com.

Note from the Authors: This article is an excerpt from Chapter 14, Section 14.2 of *Visual Basic 6 How to Program*. In this article, we discuss Visual Basic 6's directory, file and drive controls. This is an introductory-level article and readers should be familiar with Visual Basic 6 concepts. The code examples included in this article show readers examples using the DEITEL™ signature LIVE-CODE™ Approach, which presents all concepts in the context of complete working programs followed by the screen shots of the actual inputs and outputs.

14.2 DirListBox, FileListBox and DriveListBox Controls

Visual Basic provides three intrinsic controls for representing the directories, files and drives on a system. **FileListBox** visually lists the files in a directory, **DirListBox** visually represents the directory structure and **DriveListBox** visually lists the drive structure. The toolbox icons representing these controls are shown in Fig. 14.1. Note that the **CommonDialog** discussed in previous chapters can also be used to select files, directories and drives.

DirListBoxes, **FileListBoxes** and **DriveListBoxes** share many properties and methods, some of which are listed in Fig. 14.2. Note that these controls are not automatically aware of each other on a form; the programmer must write code to “tie” them together. For example, if the user selects a drive from a **DriveListBox**, the program would respond to that event by updating a corresponding **DirListBox** to display the directories from that drive. We demonstrate how this is accomplished in our first example.

When using a **DirListBox** control or a **FileListBox** control, the **Path** property sets the directory path that is displayed. The **Drive** control’s **Drive** property specifies which **Drive** is displayed. The primary event procedure of a **DirListBox** and a **DriveListBox** is **Change**—the same event procedure for **ListBoxes** and **ComboBoxes**. The primary event procedure for a **FileListBox** is **Click**.

FileListBoxes contain more properties than **DirListBoxes** and **DriveListBoxes**. Some of these additional properties are listed in Fig. 14.3.



Fig. 14.1 DirListBox, FileListBox and DriveListBox toolbox icons.

Property/method	Description
<i>Properties</i>	
DragIcon	Icon which is displayed during drag-and-drop operation.
DragMode	Integer . Automatic or manual drag-and-drop.
Enabled	Boolean . Specifies whether or not the user is allowed to interact with the control.
List	String array. Array that stores the Strings that appear in the controls.
ListCount	Integer . Number of items in the List properties array.
ListIndex	Integer . Index of selected List property item. Index begins at 0 and is -1 when a value is not selected.

Fig. 14.2 Some DirListBox, FileListBox and DriveListBox common properties and methods (part 1 of 2).

Property/method	Description
MousePointer	Integer. Specifies the shape of the mouse pointer when over the control.
Visible	Boolean. Specifies whether or not the control is visible.
<i>Methods</i>	
Drag	Starts, terminates, or aborts drag operations.
Refresh	Forces the control to repaint itself.
SetFocus	Transfers the focus to the control.

Fig. 14.2 Some **DirListBox**, **FileListBox** and **DriveListBox** common properties and methods (part 2 of 2).

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

Fig. 14.3 Some **FileListBox** properties.

The program of Fig. 14.4 allows the user to navigate the system directories. The GUI contains one **FileListBox**, one **DirListBox** and one **DriveListBox**. Selecting a drive from the **DriveListBox** updates the **DirListBox**, which updates the **FileListBox**.

```

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).

```
filFileBox.Path = dirDirBox.Path
```

in event procedure **dirDirBox_Change**. The **FileListBox** is now “tied” to the **DirListBox**. Every time the user changes the **DirListBox**’s selected directory, the **FileListBox** contents are changed to the contents of that directory.

The **DirListBox** is tied to the selected drive with the assignment

```
dirDirBox.Path = drvDriveBox.Drive
```

in event procedure **drvDriveBox_Change**. The **DirListBox** is now “tied” to the **DriveListBox**. Every time the user changes the **DriveListBox**’s selected drive, the **DirListBox** directory structure changes to the contents of that drive. Note that the **a:** drive is usually a floppy drive and that error handlers are often necessary to handle errors associated with accessing this drive.

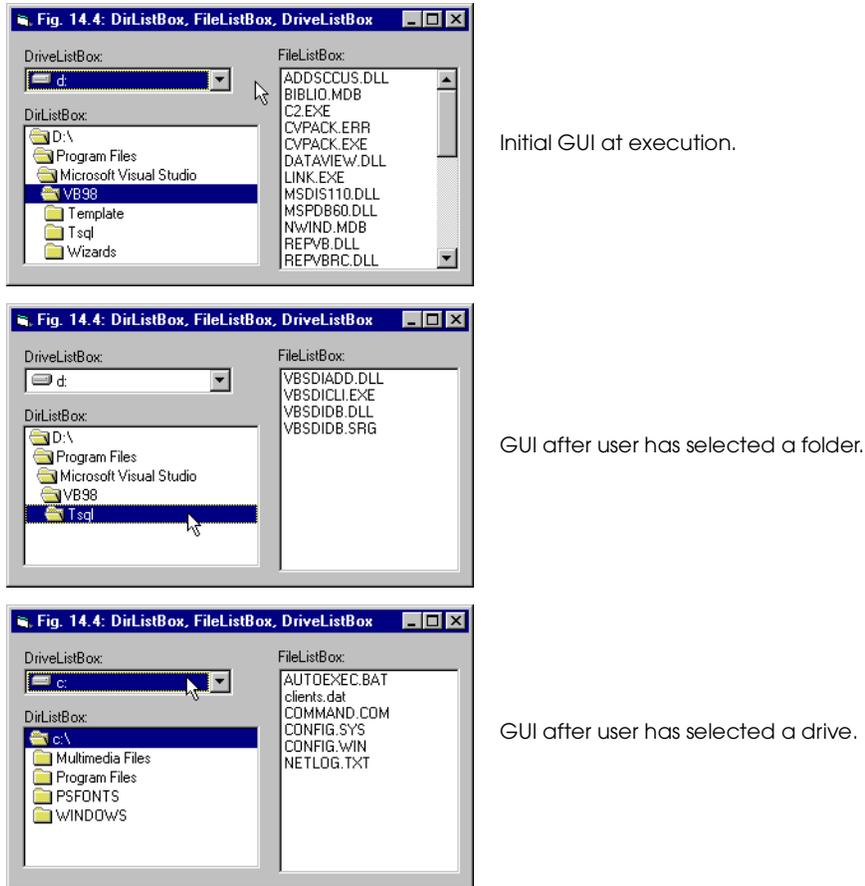


Fig. 14.4 Demonstrating controls **DirListBox**, **FileListBox** and **DriveListBox** (part 2 of 2).

Accessing the **a:** drive when it does not contain a disk is a common error. The drive is referred to as *not ready*. This specific error is trapped and dealt with, displaying a **MsgBox** dialog that gives the user the opportunity to either **Retry** or **Cancel**. By allowing the user to **Retry**, the user can ensure that a floppy is properly placed in the drive, and by allowing the user to **Cancel**, the attempt to access **a:** is canceled.



Common Programming Error 14.1

*Accessing the **a:** drive (or similar removable storage drives) when it is not ready is a runtime error.*

When **Cancel** is pressed, the line

```
drvDriveBox.Drive = drvDriveBox.List(1)
```

displays the drive at index **1** (the **a:** drive is at index **0**). Each **Drive** item can be accessed through the **List** property.

**Good Programming Practice 14.1**

Prefixing **DirListBox** controls with **dir** allows them to be identified easily.

**Good Programming Practice 14.2**

Prefixing **FileListBox** controls with **fil** allows them to be identified easily.

**Good Programming Practice 14.3**

Prefixing **DriveListBox** controls with **drv** allows them to be identified easily.