Visual Basic Tips & Tricks

VB for DOS

VB for Windows

Help Files

Release History
About Developer
About VB Tips & Tricks

Visual Basic For DOS

Files & Directories

Files & Directories

Finding Directories

Finding Directories

How do you test to see a directory exists with Visual Basic? You might want to use the DIR$ command like this:

```
DIR$("c:\test\*.*)
```

This will work as long as there are files in the directory. But how about if the directory is empty? The above code won't work.

To get around this I use the "nul" specification. Every directory has a "nul" file in it, regardless if there are any other files in it or not. Below is how to use it:

```
XY$ = DIR$("c:\test\nul")
IF XY$ <> "nul" THEN
        MKDIR "c:\test"
END IF
```

Submitted By: David McCarter

Visual Basic For Windows

Buttons & Image Control
Form_Load () Events
List Boxes
Hot Spots
Menus
Text Boxes
Miscellaneous

Buttons & Image Control

Image Control As A Button

Image Control As A Button

One of the easies techniques for adding graphical effects to your program is to use an image control as a button. When the image control receives a Click event, you simply substitute the value of the Picture property. The key to this technique is to define a pair of invisible image controls with pictures corresponding to the up and down statues of the control.

For example, you could create a button that visually represents a locked and unlocked state. One adtantage of using icon files rather than bitmap files is that any underlying image shows trough the mask area of the icon .

When the for is loaded, the Form_Load event procedure sets the appropriate image in the image control:

```
Sub Form_Load()
        Padlock.Picture = LockOpen.Picture
End Sub
```

The image control responds to the click event by replacing the picture in the control:

```
Sub Padlock_Click()
Static LockedFlag As Integer
If LockedFlag Then
        Padlock.Picture = LockOpen.Picture
        Else
                Padlock.Picture = LockClosed.Picture
End If
LockedFlag = Not LockedFlag
End Sub
```

Source: Microsoft Developer Network News, July 1993

List Boxes

Tab Stops In A List Box

Tab Stops In A List Box

The standard Visual Basic list box supports tab stops. This means that if the string value you add to the list box contains tab characters, the tabs cause the list box to display columns of strings.

Here's how to add the first item to the list box:

```
t=Chr$(9)
name="Michael Cage"
list1.AddItem name + t + "44" + t + "C/F"
```

You could also narrow the width of the list box so that you don't display the second and third columns. This techique allows you to store multiple strings per item, while only displaying the first string. Notice, however, that you would need to write additional code to extract specific strings.

Taking the idea of a list box as a storage mechanism one step further, you could make the list box invisible and only refer to it in your code.

Source: Microsoft Developer Network News, July 1993

Hot Spots

Menus

Pop-up Menus

Pop-up Menus

One of the new features of Visual Basic 3.0 is pop-up menus. You can easily create a pop-up menu from an existing menu structure. Let's look at how you would create a pop-up menu from the tradional menu in the Blanker sample application provided with Visual Basic (in your VB\SAMPLES\GRAPHICS directory).

First, use Menu Design windows to set the Visible property of mnuOption to False. Then, add the following event procedure to display the pop-up menu:

```
Sub Form_MouseUp (Button As Integer,...)
If Button = 2 Then
        PopupMenu mnuOption
End If
End Sub
```

Notice that the Form_MouseUp event procedure uses the right mouse button to display the menu.

Source: Microsoft Developer Network News, July 1993

Text Boxes

Form_Load () Events

Detecting Previous Instances Of A Program

Detecting Prevous Instances Of A Program

There are times when you may want to prevent a second instance of your program from running. The App opject provides a PrevInstance property that allows you to determine whether a previous instance of the program is running. Here's how you might write your code:

```
Sub Form_Load ()
If App.PrevInstance Then
        msg$ = App.EXEName & " already running "
        MsgBox msg$, 48
        End
End If
End Sub
```

Notice that the procedure uses the EXEName property of the App object to display the program's name in the Visual Basic message box.

Source: Microsoft Developer Network News, July 1993

Miscellaneous

Automatic Selection Of Text

Automatic Selection Of Text

When using text boxes, it is often useful to generate automatic selection of text when the control gets focus. You can do this easily by adding a couple of lines to the GotFocus even procedure:

```
Sub Text1_GotFocus ()
Text1.SelStart = 0
Text1.SelLength = 65000
End Sub
```

Notice that the length value for SelLength is 65000, nearly the maxium length allowed in a text bos. This forces Visual Basic to use the actual length of the text as the SelLength.

Source Microsoft Developer Network News, July 1993

Windows Help Files

Creating Bullets

Creating Bullets

One night I spent at least 2 hours trying to figure out how to create bullet items in a Windows 3.1 Help File. I found some Microsoft documentation which in both places it showed how to do it wrong! Below is the way that I found that works when using the QDHelp shareware program to compile the RTF file:

/para \tx360 \li360 \fi-360
{\f1 \'B7} \tab
Configurable to use many different sizes of diskettes.
Can even use 3 1/2 1.4MB (2.7MB) floppies compressed with Stacker.
/endpara

The '\tx360' RTF command sets the first tab stop to 360. The '\li360' sets the left indent to 360 and the '\fi-360' sets the first line indent to -360 or 0 in this case. The '\f1' sets the font to #1, which should be the Symbol font. The '\'B7' is the hex code for the bullet character in the Symbol font. The '\tab' moves the first line of text to the tab stop created with '\tx360'.

Here is what it looks like:

- Configurable to use many different sizes of diskettes. Can even use 3 1/2 1.4MB (2.7MB) floppies compressed with Stacker.

Submitted By: David McCarter

Release History

Version 1.0 Released on: 9/9/93 - File Name: VBTIPS10.ZIP

Developer Information

The Visual Basic Tips & Tricks Help File

Is compiled by: David McCarter

I can be reached at:

DPM Computer Solutions, 8430-D Summerdale Road, San Diego, CA 92126-5415 USA.

InterNet-MCCART@VAXD.GAT.COM.
Compuserve Users-Contact me using the address: INTERNET:MCCART@VAXD.GAT.COM

If I (or someone that I know) has tested the submitted code to make sure that it works, then we tell you by adding a graphic like this 

for the Visual Basic version it was tested with.

Disclaimer: I will try to make sure that all the coding in this help files is correct and works! I am not responsible for any damage that might happen to your computer or programs. REMEMBER...save your work before trying any coding listed here.

This Help File was written with the following Shareware programs:
Quick & Dirty Help
WinEdit

About VB Tips & Tricks

The future of this project rests on you shoulders... I wanted to develop this help file to provide some sort of centralized forum for users to share **VB Tips & Tricks**. I want to pool information from many different sources into one help file.

As you might already know, Microsoft books and help files are (in my opinion) not written very well. They do not contain much help that a normal programmer might want. Some of their information is even printed wrong! So where does one turn? To off-the-shelf books, user forums, magazine articles and the like. Try to keep all that organized to retrieve the information you need quickly. Not so easy, is it?

I want this help file to provide HELP with small **VB Tips & Tricks**. Undocumented ones, work arounds, easier way to do things etc... **But I need your help!**. I need your input!

Please submit your **VB Tips & Tricks** for others to use. Since this help file is freeware or public domain, the only thing you will receive is your name imbedded in this help file forever.

How to submit your VB Tips & Tricks:

Write down your tip or trick explaining exactly what is it is. Provide any coding (make sure it works) and graphics if you want. Text must be is ASCII format and graphics must be 16 color in a BMP format.

Send your submission to:
DPM Computer Solutions
c/o VB Tips & Tricks
8430-D Summerdale Road
San Diego, CA 92126-5415 USA

Please submit them on a 3 1/2 disk using a floppy disk mailer.

You can also save some $$$ and electronically send it to me at:
InterNet - MCCART@VAXD.GAT.COM
CompuServe - INTERNET:MCCART@VAXD.GAT.COM
Send any files (please zip them first to 500K or less) using uuencode or binhex coding.

How To Receive Updates- The easiest way is to receive them electronically through e-mail. To get on the e-mailing list just submit a VB Tip or Trick! Use my e-mail address above and send me your name, e-mail address, coding preference (uuencode or binhex) and file size limitations your system might have along with your VB Tip or Trick. My system has a 500K limit, so if and when the zipped file reaches that size, e-mail will no longer work. This will also be posted on many InterNet systems and BBS systems to be listed in the next version.