An open letter to all owners of PC Magazine's Visual Basic Programmer's Guide to the Windows API.
version 2.0!!

Thank you.

How else can I start this letter? The Visual Basic Programmer's Guide to the Windows API is about to go into its fourth printing - almost unheard of in a market where most books are lucky to sell out their first. As usual, the publisher has asked that I provide them with information on any changes or corrections that are needed so that they can be added to the next printing.

When Visual Basic 3.0 came out, I provided them with changes and corrections to make the book compatible with the new version of Visual Basic, and with their permission also posted the first version of this file - making all of those changes available to those who already have the book.

Once again, the corrections are so minor that it does not make sense for you to go out and purchase a new copy of the book. So once again, I am pleased to be able to provide a complete list of corrections and changes so that you can upgrade the text and files on your own.

This file is based on the corrections as submitted to the publisher. I can't guarantee that they will match the 4th printing exactly (they will be proofing the text for grammar and spelling), but it should be close enough.

This file may be distributed freely - please pass it around, post it on other BBS systems, etc.

Format of this file:

The first part of this file contains the changes required to bring the 3rd printing of the book to match the 4th printing. The complete contents of the original apibk.doc file follows. (I realize this increases your download time somewhat - but it seemed easier than trying to keep track of two separate versions). This latter part lists the changes from the first and second printings to the third printing. (The first and second printing were substantially identical). The first and second printings can be identified by the "Covers Visual Basic 2.0" red circle on the cover. In the third and fourth printing this circle says "Covers Visual Basic 2.0 & 3.0".

All page numbers in the first part of this document refer to the 3rd printing. If the page number is different for the second printing, the 2nd printing page number will follow in parenthesis. All page numbers in the second part refer to the first and second printing.

Final Comments:

Thanks to those who submitted suggestions and corrections for this printing. In addition to those mentioned earlier, I would like to recognize Brook Rimes and John Beckett (I swear, John must have checked it cover to cover!).

And forgive me if I make a brief plug: My company, Desaware, makes what I think are some of the finest and most unique (and generally useful) Visual Basic add-ons in the business. I encourage you to look at the demo files for our SpyWorks-VB, Custom Control Factory and CCF-Cursors products that are on the disk that came with the book.

In March 94 we will be shipping a new package called **VersionStamper-VB** which addresses the problems of distributing VB executables: detecting incompatible and obsolete VBX and DLL files at runtime on a system, and embedding windows standard version resource information into your executable. We will also be shipping our new **Common Dialog Toolkit** - an advanced application note for SpyWorks-VB that lets you take advantage of *all* of the capability of the Windows common dialog DLL.

Please feel free to give us a call, or send some Email and we'll be glad to tell you about any of these products and send you additional information.

And once again, thank you for your support.

Daniel Appleman

Desaware: (408) 377-4770, CIS: 70303,2252, Internet 70303.2252@compuserve.com

Page 41

(For some reason this fix was not placed in the 3rd printing even though it was requested).

The second line of code is:

If IsWindowVisible= -1 then ....

should be:

If IsWindowVisible(hWnd%) = -1 then ....

Page 102 - Entry for WS_CLIPSIBLINGS

WS_CLIPSIBLINGS 2000000 should be:

WS_CLIPSIBLINGS 4000000

Page 115 - DeferWindowPos - Change declaration (Add % symbol for 4 parameters)

Declare Function DeferWindowPos% Lib "User" (ByVal hWinPosInfo%, ByVal hWnd%, ByVal hWndInsertAfter%, ByVal x, ByVal y, ByVal cx, ByVal cy, ByVal wFlags%) should be:

Declare Function DeferWindowPos% Lib "User" (ByVal hWinPosInfo%, ByVal hWnd%, ByVal hWndInsertAfter%, ByVal x%, ByVal y%, ByVal cx%, ByVal cy%, ByVal wFlags%)

Page 137

(For some reason this fix was not placed in the 3rd printing even though it was requested).

SendMessage, SendMessageBynum and SendMessageBystring functions - VB Declarations section

SendMessage, SendMessageBynum and SendMessageBystring all return longs. Change the **%** sign at the end of each to **&**

SendMessageBynum and SendMessageBystring: Both should be aliased to **"SendMessage"**, not **"PostMessage"**.

Page 177 - GetAsyncKeyState

VB Declarations Declare Function Lib "User" (ByVal vKey%) should be

VB Declarations Declare Function GetAsyncKeyState Lib "User" (ByVal vKey%)

Page 307 - GetObject

In the parameter section, the table entry for hObject:

hObject Integer - Handle to a pen, brush, font, bitmap or palette. should be

hObject Integer - Handle to a pen, brush, font, bitmap.

Page 313 - MoveTo

Return Value Integer - TRUE (nonzero) on success, zero otherwise. Should be:

Return Value Long - The low word of the result contains the X coordinate in device coordinates. The high word contains the Y coordinate.

Page 485 - Table 11.1, the entry for dmSize

dmSize The Total size of the DEVMODE structure including the private data area. should be:

dmSize The Total size of the DEVMODE structure not including the private data area.

Page 486 - Table 11.1, the entry for dmPaperSize

The last sentence in the dmPaperSize entry is:

.... as defined in file APITYPES.TXT with the prefix DMPAPER_. should be

.... as defined in file APICONST.TXT with the prefix DMPAPER_.

Page 487, Escape function - Return Value

The first sentence under the Return Value section is:

Integer - Depends on the nEscape parameter. Unless otherwise specified in table 11.2, the return is zero on success, and a negative number on error based on the following constants: should be:

Integer - Depends on the nEscape parameter. Unless otherwise specified in table 11.2, the return is greater than zero on success, zero if the Escape is not implemented, and a negative number on error based on the following constants:

Page 490 - Table 11.2 GETSETPRINTORIENT

Change this entry to read as follows:

GETSETPRINTORIENT	**lpInData** points to a 20-byte structure. The first 32 bit LONG value contains the orientation. If **lpInData** is NULL, the Escape function will return the current orientation. The **DeviceCapabilities** and **ExtDeviceMode** functions make this escape obsolete.

Page 491 - Table 11.2 - QUERYESCAPESUPPORT

Change **QUERYESCAPESUPPORT** to **QUERYESCSUPPORT**

Page 512

The block of code at the bottom of the page should be changed as follows:

```
' Get a copy of the DEVMODE structure for this printer
' First find out how big the DEVMODE structure is
bufsize% = agExtDeviceMode%(hWnd, libhnd%, 0, devname$, devoutput$, ⇔
agGetAddressForObject(dm), 0, 0)
' Allocate a buffer of that size and get a pointer to it
dminstring$ = String$(bufsize%, 0)
dminaddr& = agGetAddressForVBString&(dminstring$)
dmoutstring$ = String$(bufsize%, 0)
dmoutaddr& = agGetAddressForVBString&(dmoutstring$)

' Get the output DEVMODE structure.
di% = agExtDeviceMode(hWnd, libhnd%, dmoutaddr&, devname$, devoutput$, dminaddr&, 0, ⇔
DM_OUT_BUFFER)

' Copy the data buffer into the DEVMODE structure
agCopyDataBynum dmoutaddr&, agGetAddressForObject&(dm), 68
' Set the orientation, and set the dmField flag so that
' the function will know that it is valid.
```

Page 518, Table 11.4

Insert the following table entry after the DC_PAPERS entry

DC_PAPERSIZE	**lpszOutput** is a pointer to an array of POINTAPI structures which are loaded with the dimensions of supported paper sizes tenths of a millimeter. Sizes are always returned for portrait mode regardless of the current printer configuration.

Page 521, Table 11.5. Replace corresponding table entry with the following

DM_IN_BUFFER	The DEVMODE structure referenced by the lpdmInput buffer will be used to set the printer driver. Only those fields that are specified by the dmFields field of the structure will be used. The settings specified by the lpdmInput buffer when this flag is set will effect the printer driver or the lpdmOutput buffer depending on the settings of the other flags. For example: Setting the DM_IN_BUFFER and DM_OUT_DEFAULT flags lets you use the lpdmInput buffer to set the configuration of the default printer.

Page 525 - StartPage function

Change the return value as follows:

Return Value **Integer-TRUE (nonzero) on success, zero otherwise.** should be:

Return Value **Integer - Value >= zero on success, negative on error.**

Page 550

The first parameter for function FindExecutable should be **lpszFile$**, not **lpszFile%.**

Page 561 - IsBadHugeReadPtr

Return Value **Integer - TRUE (nonzero) if the specified memory block is valid and readable by this application** should be:

Return Value **Integer - TRUE (nonzero) if the specified memory block is invalid and not readable by this application.**

Page 561 - IsBadHugeWritePtr

Return Value **Integer - TRUE (nonzero) if the specified memory block is valid and writable by this application** should be:

Return Value **Integer - TRUE (nonzero) if the specified memory block is invalid and not writable by this application.**

Page 561 - IsBadReadPtr

Return Value **Integer - TRUE (nonzero) if the specified memory block is valid and readable by this application** should be:

Return Value **Integer - TRUE (nonzero) if the specified memory block is invalid and not readable by this application.**

Page 561 - IsBadStringPtr

Return Value **Integer - TRUE (nonzero) if the specified memory block is valid and contains a valid null terminated string.** should be:

Return Value **Integer - TRUE (nonzero) if the specified memory block is invalid and does not contain a valid null terminated string.**

Page 561 - IsBadWritePtr

Return Value **Integer - TRUE (nonzero) if the specified memory block is valid and writable by this application** should be:

Return Value **Integer - TRUE (nonzero) if the specified memory block is invalid and not writable by this application.**

Page 566 - Declaration for WinExec

Declare Function WinExec% Lib "Kernel" (ByVal lpCmdLine$, nCmdShow%) should be

Declare Function WinExec% Lib "Kernel" (ByVal lpCmdLine$, ByVal nCmdShow%)

Page 567 - Top of page:

Return Value **Integer - Greater than 32 on success**..... should be

Return Value **Integer - Module handle for the executed application - value > 32 on success**....

Page 870

(For some reason this fix was not placed in the 3rd printing even though it was requested).

Documentation is missing for the agVBSetControlFlags function.

agVBSetControlFlags

VB Declaration:

Declare Function agVBSetControlFlags& Lib "Apiguide.dll" (ctl As Control, ByVal mask&, ByVal value&)

Description:

This function is used to control the palette status of a control and returns the current status of the control.

Use with VB:

Can be used to specify or determine when a control is palette, and whether or not it currently owns a palette. In practice, this is only effective for determining status. You can use this function to set the palette awareness of a control only if you take over all aspects of selecting and realizing palettes. This requires a subclassing tool capable of detecting both the windows palette messages and the internal Visual Basic palette messages.

Parameters:

ctl - A control or form

mask - Set a bit in the mask to 1 to indicate that it should be changed according to the value parameter.

value - Indicates the new value for the bits specified by the mask parameter.

Bit 0 is set to 1 to indicate that the control owns a palette.
Bit 1 is set to 1 to indicate that the control is palette aware.

Returns Value - Long - A value describing the current state of the control.

Page 794 - Last line in Sub UpdateDisplayLine()

LabelShowLine.Caption = linebuf$ should be

LabelShowLine.Caption = Left$(linebuf$, lc%)

Page 923 - Change declaration (add % symbol after name and for 4 parameters)

Declare Function DeferWindowPos Lib "User" (ByVal hWinPosInfo%, ByVal hWnd%, ByVal hWndInsertAfter%, ByVal x, ByVal y, ByVal cx, ByVal cy, ByVal wFlags%) should be:

Declare Function DeferWindowPos% Lib "User" (ByVal hWinPosInfo%, ByVal hWnd%, ByVal hWndInsertAfter%, ByVal x%, ByVal y%, ByVal cx%, ByVal cy%, ByVal wFlags%)

Page 926 - Declarations for Escape, EscapeBynum and EscapeByString

For all three of these declarations, the fourth parameter is currently listed as **lplnData**. This should be **lpInData**.

Page 927

The first parameter for function FindExecutable should be **lpszFile$**, not **lpszFile%.**

Page 952

Change the reference to **shellapi.dll** in function ShellExecute and ShellExecuteBynum to **shell.dll.**

Page 955 - Change declaration (add ByVal to last parameter)

Declare Function WinExec% Lib "Kernel" (ByVal lpCmdLine$, nCmdShow%) should be:

Declare Function WinExec% Lib "Kernel" (ByVal lpCmdLine$, ByVal nCmdShow%)

File Changes

Apidecs.bas and Apidecs.txt

Add the following two declarations to the file:

Declare Function FindWindowByClass% Lib "User" Alias "FindWindow" (Byval lpClassName$, Byval lpWindowName&)

Declare Function FindWindowByWindow% Lib "User" Alias "FindWindow" (Byval lpClassName&, Byval lpWindowName$)

Correct the declaration for DeferWindowPos as follows:

Declare Function DeferWindowPos% Lib "User" (ByVal hWinPosInfo%, ByVal hWnd%, ByVal hWndInsertAfter%, ByVal x%, ByVal y%, ByVal cx%, ByVal cy%, ByVal wFlags%)

Correct the declaration for the WinExec function as follows:

Declare Function WinExec% Lib "Kernel" (ByVal lpCmdLine$, ByVal nCmdShow%)

For the three declarations for the **Escape** function, the fourth parameter is **lplnData** should be **lpInData**. If you change this, be sure to check if you need to change any of your applications that use this function.

ApiConst.Txt and ApiConst.Txt

Add the following lines to the file:

" SendMessage Flag

Global Const HWND_BROADCAST = -1

The following is the original apibk.doc file that describes how to upgrade the first or second printing of the Visual Basic Programmer's Guide to the Windows API to correspond to the 3rd printing. It also discusses issues relating to compatibility with Visual Basic 3.0

The release of Visual Basic 3.0 coincides with the release of the third printing of this book. It won't surprise you to know that this printing has been revised for the new release of VB. However, since the book is essentially a Windows SDK for Visual Basic programmers, the changes for version 3.0 were quite minor (this would not be the case had we been dealing with a major release of Windows).

In fact, the changes are so minor that it makes no sense for you to go out and buy a new copy of the book. Instead, as a public service and for the sake of good customer relations, this file contains a list of all of the changes and corrections that were incorporated into the third printing.

This file is based on the corrections as submitted to the publisher. I can't guarantee that they match the 3rd printing exactly (they will be proofing the text for grammar and spelling), but it should be close enough. On the other hand, because of lead times in the publication process, this document contains corrections that are not present in the 3rd printing (those corrections will also appear in the readme.txt file on the disk that comes with the second printing - so there is no need to pass this document on to people who have the 3rd printing).

This file may be distributed freely - please pass it around, post it on other BBS systems, etc.

Thanks to all of you who spotted errors in the book and forwarded them to me. Unfortunately, I did not keep track of who submitted suggestions and corrections, but off the top of my head I do remember Jonathan Zuck, Keith Pleas and Ted Young. To the others - I apologize for my forgetfulness (feel free to drop me a note and I'll be sure to mention you in the next set of changes).

Also, allow me to take this moment to encourage you to take a look at the SpyWorks-VB and CCF-Cursors demo program that comes on the book's disk. SpyWorks-VB is especially useful in conjunction with the book when it comes to taking advantage of advanced Windows API techniques.

Thank you for your support.

Daniel Appleman

Page xxiii

Header **Visual Basic Compatibility** has been renamed **Visual Basic Version Compatibility**

The last two paragraphs on the page have been replaced with the following text:

> The original printing of this book was released for version 2.0 of Visual Basic. This edition has been revised as needed to be compatible with both Visual Basic 2.0 and Visual Basic 3.0. The changes were quite minor, owing to the fact that while Visual Basic changed, the underlying Windows API has not. All of the sample files and listings have been tested with Visual Basic 3.0, however they are still presented in version 2.0 format in order to ensure compatibility with both versions.
>
> Some discussion of Visual Basic 1.0 has been left in this revision where appropriate. The listing for the RectPlay example in chapter 4 discusses how to interpret listings for version 1.0 if necessary.

Page xxiv

The following sentence is added after the first sentence in the *Closing Notes* section.

> I never imagined that it would be as well received by the Visual Basic community as it has been.

Page 8, 2nd paragraph from the bottom, 4th line

> Change C/C++ 7.0 to Visual C++

Page 23

Table 2.3, footnote #2

Change **VB 2.0 implements...** to **VB 2.0 and later implements...**

Page 24

In Sub Command1_Click:

Change **For x% = 1 to 50** to **For x% = 1 to 1000**

This change still makes the same point, but the difference in performance will be more noticeable than it would be with only 50 entries.

Page 41

The second line of code is:

If IsWindowVisible= -1 then ....

should be:

If IsWindowVisible(hWnd%) = -1 then ....

Page 45

Bottom paragraph, 3rd line.

Change **376836** to **6029316**

Page 46

Table 3.4 should be changed as follows:

Determining Bit Values in a Number (Example &H805C0004) should be
Determining Bit Values in a Number (Example &H05C0004).

Change line 5 as follows:

5 5 20-23 22 and 21 should be
5 5 20-23 22 and 20

Page 56

2nd paragraph from the bottom, 2nd line.

Change **form of a Visual Basic 2.0 module...** to **form of a Visual Basic module....**

Page 69

RectPlay Program Listings section, 2nd paragraph, first line, should be:

Program listings appear as saved in Visual Basic version 2.0 ASCII file format which is also compatible with Visual Basic version 3.0.

Page 70

2nd paragraph, 1st line

Change **version 2.0 of Visual Basic** to **version 2.0 and 3.0 of Visual Basic**

Page 80

Add the following entry to table 4.5

GetUpdateRect Determines the portion of a window that needs to be updated

Page 137

SendMessage, SendMessageBynum and SendMessageBystring functions - VB Declarations section

SendMessage, SendMessageBynum and SendMessageBystring all return longs. Change the **%** sign at the end of each to **&**

SendMessageBynum and SendMessageBystring: Both should be aliased to **"SendMessage"**, not **"PostMessage"**.

Page 195

The correct declaration for SystemParametersInfoByval is:

Declare Function SystemParametersInfoByval% Lib "User" Alias "SystemParametersInfo" (ByVal uAction%, ByVal uParam%, ByVal lpvParam As Any, ByVal fuWinIni%)

Page 219

2nd paragraph from the bottom, 5th line.

Change **... of how many display pixels are in an inch is made ...** to
... of how many display pixels are in a logical inch is made ...

Page 243

Add the following entry into the RASTERCAPS table entry:

RC_FLOODFILL: FloodFill API function is supported.

Page 268

In table 7.2 change **GetObject** to **GetObjectAPI**

(Note: The keyword **GetObject** became a reserved word in Visual Basic 3.0. In order to prevent conflict, the book and all sample and declaration files has been modified to use **GetObjectAPI** as an alias for the **GetObject** API function in much the way that **SetFocusAPI** is an alias for the **SetFocus** API function.)

Page 307

Change the declaration for **GetObject** to:

GetObjectAPI

Declare Function GetObjectAPI% Lib "GDI" Alias "GetObject" (ByVal hObject%, ByVal nCount%, ByVal lpObject&)

Page 320

In the "Use with VB" section for the SetROP2 function, nDrawMode parameter description, change the reference to table **7.9** to **7.8**.

Listing 8.4 (page 341)

Last line on the page - change **GetObject** to **GetObjectAPI**

List 8.9 (pages 349 through 358)

Change all references to **GetObject** to **GetObjectAPI**.

Change all references to **Update** to **DoUpdate** (*Update became a reserved word in VB 2.0 - the 1st printing disk was corrected, but it did not get into the book. This change can be found in the 1st printing readme.txt file).*

The indentation in this listing does not match the conventions used in the other sample programs. How this happened is still somewhat of a mystery to me - but with luck it will be fixed for this printing.

Page 367

Change all references to **shellapi.dll** in the ExtractIcon function declaration and description to **shell.dll**.

Page 379

1st paragraph, delete the 3rd sentence and add:

You will learn how to create custom checkmarks for checked menus, and how to use any bitmap as a menu entry in place of a string. You will also learn how to customize floating popup menus that can appear anywhere on the screen.

Page 387

In the section **Tracked Popup Menus** delete the first sentence and add:

> Visual Basic 3.0 provides direct support for floating popup menus to appear anywhere on the screen using the **PopupMenu** command. The **TrackPopupMenu** API function can also be used to create popup menus in cases where customization is required or for use with previous versions of Visual Basic.

Page 388

Insert a new subheading as follows:

> **Menus, System Menus and Subclassing**
>
> Subclassing is a technique which allows you to intercept Windows messages going to a form. This technique can be used to detect the WM_COMMAND Windows message directly, eliminating the need to ensure compatibility with a Visual Basic menu structure when using menu API functions. It also allows you to intercept the WM_SYSCOMMAND message which makes it practical to customize an application's system menu. Refer to the Message Handling section in chapter 17 for more information on subclassing and the tools required to use this powerful technique.

Listing 9.4 (pages 397-405)

Change all references to **GetObject** to **GetObjectAPI**.

Once again, the indentation does not consistantly follow Basic standards.

Page 397, function GetFlagString$

Change the 3rd through 4th lines in the function to the following:

```
If (menuflags% And MF_CHECKED) <>0 Then
    f$ = f$ + "Checked"
Else
    f$ = f$ + "UnChecked"
```

Page 410, 412, 413,414

EnableMenuItem function - **wIDEnableItem** parameter
GetMenuState - **wID** parameter
GetMenuString - **wIDItem** parameter
HiliteMenuItem - **wIDHiliteItem** paramater

In each of these cases, the first line says: "Identifier of the menu entry to check or uncheck". Modifiy this to match the description of each of these functions. This proves once and for all that the benifits of incorporating the "cut" and "Paste" operation into word processors is not without its drawbacks. Thanks to Ted Young for spotting this one.

Page 420

Use with VB section - add before the first sentence:

Visual Basic 3.0 provides direct support for tracked popup menus, however this function remains useful for customized menus and use with earlier versions of Visual Basic.

Listing 10.8

Change all references to **GetObject** to **GetObjectAPI**.

Listing 11.6

Change all references to **GetObject** to **GetObjectAPI**.

Page 545

> Listing 12.4 heading should be: **Project Listing File EXECDEMO.MAK**
>
> Listing 12.5 heading should be: **Form Description for File EXECDEMO.FRM**

Listing 12.6

Correct indentation for Sub File1_Click( )

Page 550

The first parameter for function FindExecutable should be **lpszFile$**, not **lpszFile%.**

The library declarations should be **shell.dll** not **shellapi.dll**.

Page 565

Change the library declaration for function ShellExecute to refer to function **shell.dll** instead of **shellapi.dll**.

Page 609

Under the Return Value section for the OpenFile function, the final sentence should read:

Errors are listed in Table 13.9 earlier in this chapter.

Page 610

The following table entry should be added to Tabel 13.12 before the OF_WRITE entry:

OF_VERIFY	Returns HFILE_ERROR if the time and date of the file specified by the lpFileName$ parameter does not match that specified by the lpReOpenBuff parameter.

Page 663

In table 15.1, change the second reference to **CF_TEXT** to **CF_TIFF**.

Chapter 16

Palettes seem to work the same under VB 2.0 and 3.0. The text was clarify to indicate this. Specifically - all indications of Visual Basic 2.0 have been changed to 2.0 & 3.0 as follows:

Page 691, par 2, line 3
Page 697, par 2, line 3
Page 698, section **Using Palettee Functions with Visual Basic** line 1
Page 710, 2nd paragraph from the bottom, last line.
Page 731, Function RealizePalette, Use with VB section line 1
Page 733, Function SelectPalette, Use with VB section line 1
Page 735, Function SelectPalette, Use with VB section line 1

Chapter 16, Function reference section

For the following functions: **DrageAcceptFiles, DragFinish, DragQueryFile** & **DragQueryPoint** change the declaration reference from **"shellapi.dll"** to **"shell.dll"**.

Page 711

In section Dragging files, line1, change **SHELLAPI.DLL** to **SHELL.DLL**.

Page 765

Add the following Comments section to the WM_MENUSELECT command.

Comments:

When a menu is closed, wParam will be zero and the low word of lParam will be &Hffff.

Page 801

EM_GETPASSWORDCHAR function, Use with VB section. Change reference to **Visual Basic 2.0** to be **Visual Basic 2.0 & 3.0**

Page 802

EM_LIMITTEXT function, Use with VB section. Change reference to **Visual Basic 2.0** to be **Visual Basic 2.0 & 3.0**

Page 805

EM_SETPASSWORDCHAR function, Use with VB section. Change reference to **Visual Basic 2.0** to be **Visual Basic 2.0 & 3.0**

Page 865

Use with VB section of agGetControlHwnd function. Change reference to **Visual Basic 2.0** to **Visual Basic 2.0 and later**.

Page 870

Documentation is missing for the agVBSetControlFlags function.

agVBSetControlFlags

VB Declaration:

Declare Function agVBSetControlFlags& Lib "Apiguide.dll" (ctl As Control, ByVal mask&, ByVal value&)

Description:

This function is used to control the palette status of a control and returns the current status of the control.

Use with VB:

Can be used to specify or determine when a control is palette, and whether or not it currently owns a palette. In practice, this is only effective for determining status. You can use this function to set the palette awareness of a control only if you take over all aspects of selecting and realizing palettes. This requires a subclassing tool capable of detecting both the windows palette messages and the internal Visual Basic palette messages.

Parameters:

ctl - A control or form

mask - Set a bit in the mask to 1 to indicate that it should be chagned according to the value parameter.

value - Indicates the new value for the bits specified by the mask parameter.

Bit 0 is set to 1 to indicate that the control owns a palette.
Bit 1 is set to 1 to indicate that the control is palette aware.

Returns Value - Long - A value describing the current state of the control.

Page 924

Change all references to **shellapi.dll** in functions DragAcceptFiles, DragFinish, DragQueryFile and DragQueryPoint to **shell.dll**.

Page 927

The first parameter for function FindExecutable should be **lpszFile$**, not **lpszFile%.**

Change the reference to **shellapi.dll** in functions ExtractIcon and FindExecutable to **shell.dll.**

Page 952

Change the reference to **shellapi.dll** in function ShellExecute and ShellExecuteBynum to **shell.dll.**

Page 933

Change the declaration of **GetObject** to:

GetObjectAPI 7 Declare Function GetObjectAPI% Lib "GDI" Alias "GetObject" ....

Page 988

Change **BitBit** to **BitBlt**

Page 1000

Change **GetObject** to **GetObjectAPI**

Page 1001

Add a reference to page **80** to function **GetUpdateRect**.

Page 1003

The following commands:

Istrcat, Istrcmp, Istrcmpi, Istrcpy, and Istrlen should be

lstrcat, lstrcmp, lstrcmpi, lstrcpy and lstrlen (lower case 'L' as the first character)

Page 1014

Change entry **SHELLAPI.DLL** to **SHELL.DLL**

Page 1015

Add the following references to the entry for **subclassing**: 388, 741-744.

Page 1020

Yep - part of the index is missing. Here are the missing entries:

APICONST.TXT

The following constants were added:

```
" SendMessage Flag
Global Const HWND_BROADCAST = -1
" Network Connection errors
Global Const WN_NOT_CONNECTED = &H0030
Global Const WN_OPEN_FILES = &H0031
Global Const WN_BAD_NETNAME    = &H0032
Global Const WN_BAD_LOCALNAME = &H0033
Global Const WN_ALREADY_CONNECTED   = &H0034
Global Const WN_DEVICE_ERROR = &H0035
Global Const WN_CONNECTION_CLOSED     = &H0036
```

Add the following after the "SetWindowPos flags" section:

```
Global Const SWP_NOSENDCHANGING = &H400
Global Const SWP_DEFERERASE = &H2000

" SetWindowPos() hwndInsertAfter values
Global Const HWND_TOP = 0
Global Const HWND_BOTTOM = 1
Global Const HWND_TOPMOST = -1
Global Const HWND_NOTOPMOST = -2
```

Apiguide.bas

Add the following declaration:

```
Declare Function agVBSetControlFlags& Lib "Apiguide.dll" (ctl As Control, ByVal mask&, ByVal
value&)
```

Sample Code Changes

The following code changes are listed by file and line number. You may wish to also change the appropriate listing in the book. Other minor changes are listed in the update instructions earlier in this document.

GetObject

Visual Basic 3.0 uses **GetObject** as a reserved word. In order to accomodate this, the declaration for GetObject has been changed to GetObjectAPI which is aliased to GetObject. The new declaration of GetObject as seen in file apidecs.txt, apidecs.bas and in Appendix E page 933 is:

Declare Function GetObjectAPI% Lib "GDI" Alias "GetObject" (ByVal hObject%, ByVal nCount%, ByVal lpObject&)

The command **GetObject** must be changed to **GetObjectAPI** in the following files:

Menulook.frm, lines 285, 391
Puzzle.frm, line 77
Picprint.frm, line 335
Stockbms.frm, line 128
Textdemo.frm, line 116