

10Tec version of vbAccelerator ImageList

What's New

v2.50 build 0008 | 2025-Nov-18

1. [Fixed] The controls on the Images property page now show captions correctly without clipping or word wrap issues at non-standard display scales (125%, 150%, etc.).
2. [Fixed] The values of the **vKeySrc** and **vKeyDst** parameters in the **SwapOrCopyImage** method were mixed up.
3. [Fixed] The **AddFromPictureBox** function of the **vbalImageList** control returned an incorrect value of 2 if the image was added successfully. Now it returns -1 if the image was added and 0 otherwise (the numeric equivalents of the Boolean True and False values respectively). This matches the functionality of the similar functions **AddFromFile**, **AddFromPNG**, and **AddFromResourceID** returning the same numeric results.

v2.50 build 0004 | 2017-Jun-16

1. [New] This release implements the ability to upload PNG files into the ImageList control. This can be done using the Add button in the corresponding property page at design time or using the new **AddFromPNG** method at run time:

```
Function AddFromPNG( _  
    ByVal sFileName As String, _  
    Optional ByVal sKey As String, _  
    Optional ByVal vKeyAfter As Variant _  
    ) As Long
```

The new functionality works only in Windows Vista or later version of the OS as it is based on the corresponding Windows API features introduced in Windows Vista.

2. [Enhancement] All design-time members related to internal infrastructure have been hidden and no longer appear in the Property Browser or IntelliSense list. These are the **DesignTimeSetKeysFromFileNames**, **DesignTimeLastFileAddPath**, **DesignTimeLastSaveLoadPath** properties and the **DesignTimeUpdateControl** method.
3. [Enhancement] At design time, the control is resized according to the current system DPI to support high-resolution monitors.

v2.40 build 0014 | 2011-Apr-15

1. [New] This release of the ImageList OCX implements a new method you can use to draw your icons with the specified transparency level. It is **DrawImageAlpha**, and it has the following syntax:

```
Public Sub DrawImageAlpha( _  
    ByVal vKey As Variant, _  
    ByVal hdc As Long, _  
    ByVal xPixels As Integer, _  
    ByVal yPixels As Integer, _  
    ByVal btAlpha As Byte _  
    )
```

The first 4 parameters are the same as in the **DrawImage** method, but the last **btAlpha** parameter is used to specify the transparency of the drawn image. It's a Byte value, and its 0 value means fully transparent image, 255 – fully opaque.

2. [Fixed] The following bug specific for MS Office VBA environment only was fixed. When you tried to open ImageList's property pages using the ellipsis button for the special property named "(Custom)" in the Property Grid, you got the Error 97 "Cannot call friend function on object which is not an instance of

defining class". You might also get this error while making changes to the image list (adding, removing icons, etc).

The source of this bug is deep inside the VBA and it should not be so. Internally we use some friend helper members to support the internal infrastructure, and this architecture works without any problems in Visual Basic 6. However, it fails in MS Office VBA by an unknown reason.

To overcome this problem, we made those friend members public. These are the members with names starting with "DesignTime": the **DesignTimeSetKeysFromFileNames**, **DesignTimeLastFileAddPath**, **DesignTimeLastSaveLoadPath** properties and the **DesignTimeUpdateControl** method. Most likely, these members are useless in your code, so simply ignore all ImageList members that start with "DesignTime".

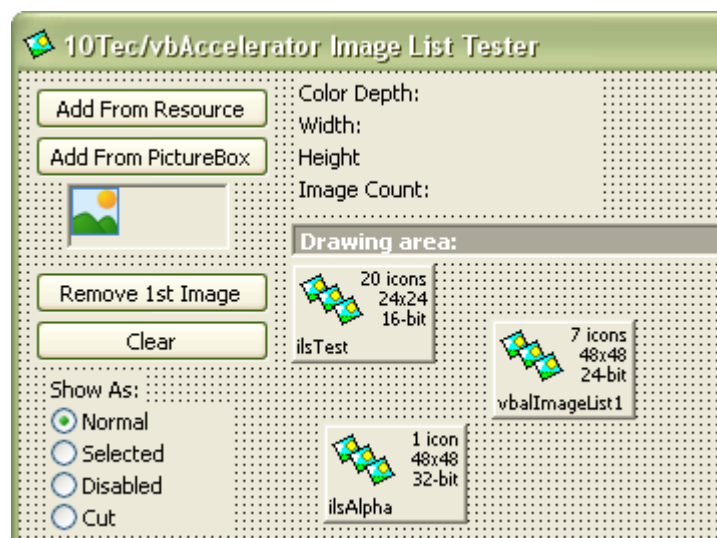
3. [Removed] The **DrawImage** method no longer has the almost useless **hExternalImI** parameter you could use to draw images from another WinAPI image list using its known system handle.
4. [Enhancement] When you added this ImageList control to the Toolbox in MS Office VBA form editor, you saw the string "vbalImI/10Tec.vbalImageList" for this ImageList control in the list of available ActiveX controls ("N" and "M" are major and minor version numbers). Now you see a more descriptive label "10Tec/vbAccelerator ImageList *N.M.M*" instead.

v2.30 build 0008 | 2010-Jul-31

1. [Important Bug Fix] The previous builds of ImageList has a serious icon degradation problem for 32-bit icons: the more you save/restore these icons using the property dialog, the darker they become. This build of control is free from this problem as its icon serialization/deserialization code used to save/restore loaded icons was rewritten from scratch using another approach.

However, because of the new serialization process, you cannot interchange icons between the previous versions of the control and this one, i.e. the property dialog's Load button and the **LoadFromFile** method cannot be used to upload icons from the .iml files saved in the previous versions. The Load button and the **LoadFromFile** method also reset the current image list and reload all images instead of adding them to the end like in the previous versions (the **ImageWidth/ImageHeight** and ColorDepth properties are also set to the actual values saved with the images in the .iml file).

2. [Enhancement] The property dialog has a new combo box called "Set size to" used to set some standard icon sizes quickly. It replaced the buttons "16", "24", "32", "48" from the previous version, and it has a new size entry "64x64".
3. [Enhancement] The control size on the form at design time is 72x50 now instead of 48x48 in the previous version. The enlarged area is used to display more info about the contained images - their quantity, size and color depth. The increased size also allows you to see more characters from long control names displayed at the bottom of the control area:



4. [Enhancement] The height of the image preview area in the property page was increased to make it possible to see images of height up to 64 pixels without clipping.

5. [New] A new **ApplyNewImageSettings** function was added. If you assigned a new value to at least one of the properties **ImageWidth**, **ImageHeight** or **ColorDepth** in your code to change the stored icons, your changes aren't applied immediately to avoid a series of changes each of them affects the icons quality. Instead of that you should call the new **ApplyNewImageSettings** function for that to apply your changes at one go and minimize the cumulative effect of icons change after setting each parameter.

The result of the **ApplyNewImageSettings** function is a Boolean value which indicates whether the changes were successfully applied. If the changes weren't applied, the function returns False and all the existing icons remain untouched.

Note that at design time you can change the icon size and/or color depth only in the property dialog (using the corresponding controls and pressing the Apply button). The changes to these 3 properties in the Property Editor have no effect because of the reason described above.

6. [Enhancement] The missing descriptions of the control members used in the Object Browser and IntelliSense were added. Some outdated descriptions were also corrected.
7. [Fixed] Some minor problems with setting and saving/restoring image keys were solved.

v2.20 build 0022 | 2010-Feb-05

1. [Change] Important! The same base numeric index 1 was set for all methods and properties which work with icon indices because of the inconsistent usage of 0 and 1 for that purpose in some previous builds. Now all icons are indexed from 1 like in the standard VB ImageList Control from the MS Windows Common Controls pack.
2. [New] This version of vbAccelerator ImageList implements the IObjectSafety COM interface which enables the use of the control without safety notification problems in such ActiveX host environments as MS Office. This interface is used by many well-known apps to detect whether an ActiveX control is safe for scripting (among of them are Internet Explorer, MS Word and Excel). For instance, MS Excel displays a warning box if an ActiveX control is not safety ("This application is about to initialize ActiveX controls that might be unsafe..."), and you should confirm each time that the control is safety when you launch your app by clicking OK in that in that message box. Now vbAccelerator ImageList implements IObjectSafety with proper settings to avoid these annoying messages.
3. [New] At design time the icon of the control on the form displays its name in addition to the number of icons it contains to make it easier distinguish different ImageList controls on the form. The size of the control's icon was also enlarged to 48 pixels to display more letters of the control's name.
4. [New][Enhanced] The property page dialog was redesigned to allow you to perform all operations with images faster and easier. The two old property pages, general image properties and the image list itself, were combined into one page which is opened by default. Now you perform all the operations in one property page opened by default, and you do not need to switch from one page to another every time.
5. [New] You can save the entire image list and load it later into this or another image list using two new buttons, Save and Load, in the property dialog page. The Save button can be used to store the current image list into a file of a special format. The Load button can be used to upload the saved image list from the file of that format; the images are added to the existing images. Using this functionality, you can even copy images between different image list controls.
6. [New][Change] Two new properties used to specify the transparent color when loading bitmap images were introduced (these are all images except icons, cursors and animated cursors). The first of them is **UseMaskColor** of a new **eilMaskColor** enumeration data type. It can accept one of the following values: **eilMaskColorNone**, **eilMaskColorTopLeftPixel**, **eilMaskColorSpecified**. The first value, **eilMaskColorNone**, is used if you wish to upload a bitmap "as is" without changing some of its pixels to transparent and it is the default setting in the ImageList control. **eilMaskColorTopLeftPixel** is used if you want to use the color of the top-left pixel as the transparent color. The last value, **eilMaskColorSpecified**, is used when you need to specify the color used as transparent. In this case the value of the other new property, **MaskColor**, is used for this purpose.

Note that you could use the functionality of the **eilMaskColorTopLeftPixel** and **eilMaskColorSpecified** options earlier through the corresponding settings in the property dialog in the

previous versions of the control, but the **eilMaskColorNone** option is an important and totally new feature not available in earlier versions.

One more change in the syntax of the **Add*** methods was made because of this enhancement. Now such methods as **AddFromHandle** do not have the **IBackColor** optional parameter as its functionality is performed through the new **UseMaskColor/MaskColor** properties you should set to the required values before you call one the **Add*** methods.

Pay attention to the fact that in contrast to the previous versions, these settings are edited in the property dialog using the same Add button options control group, and they are saved between the sessions of work with the property page dialog. They are saved as the normal properties of the control which can be also viewed and edited in the standard Property Editor. This functionality is very useful if you add images using the same transparent color options many times (for instance, when debugging your application or experimenting with different color options) – in the previous versions they were lost each time when you closed the property dialog, and you needed to set them again and again when you open the property page.

7. [Change] The "Set Keys from Filenames" option in the property page dialog is saved before sessions of work. This check box is now set by default (when you first place the control on a form) because in the general case, to have robust code, the images in an image list should be accessed by keys, and we advise that you do that by specifying the required image keys directly in their file names. If you do so, you will have your images uploaded into an image list as required with minimal efforts from your side.
8. [Enhanced] The property page dialog "remembers" the last folder you used to add images with the Add button, and this folder is opened automatically the next time you click the Add button instead of a default folder. The folder is remembered even if you close the form with the control or the entire IDE. This feature is very handy if you add images from the same folder several times, and now you do not need to waste your time searching for this folder every time when you add a new portion of images.

The same functionality was implemented for the Save\Load buttons, but they have their own folder setting.

9. [Enhanced][Change] vbAccelerator ImageList no longer uses the **ILC_COLOR** color depth setting (it was also removed from the **eilColorDepth** enumeration). It was used in the previous versions as the default color depth setting, but it was useless as it means the default system color depth setting treated as 4-bit color on all modern systems. Now the control uses the **ILC_COLOR24** setting by default suitable for the most practical cases.
10. [Enhanced] One annoying thing with the Add dialog when you select multiple files was fixed. In the original version, the files were added not in the same order as they were placed in the "File name:" text box in the "Choose Icon to Add" dialog.

To tell the truth, that dialog is the standard Windows File Open dialog, and it sorts the file names not in the same order in which you selected them with the mouse (and it does that using a strange algorithm). But our version adds the icons at least in the same order in which the file names are placed in the "File name:" text box in that dialog.

11. [Enhanced] The Add button added a new image(s) before the selected one, but now it works in a more natural way – it adds the new images after the selected item.
12. [Enhanced] When you select images using the "Select Previous"/"Select Next" graphical buttons next to the selected image index field, the control automatically scrolls the image list if required to make the new selected image visible.
13. [Enhanced][Fixed] The drawing of the images on the control's property page was polished (all parts of the special rectangle around the selected image is drawn without clipping in all cases, the properties of the horizontal scroll bar are determined more precisely, etc).
14. [Enhanced] Now you can add JPEG files and filter them as a separate category of files in the Open File dialog opened by the Add button in the property page.
15. [Enhanced] The Remove button now displays a simple Yes/No confirmation dialog before the selected image is really deleted.
16. [Change] The type of the **ItemKey** property was changed from Variant to String as it works only with strings.

17. [Change] In the property dialog, if you set the "Set Keys from Filenames" option, the names of the files are used "as is" as the image keys without case conversion (they were converted to uppercase in the previous versions).
18. [Change] To avoid any confusion with the use of the words "colour" and "color" you might see in different parts of the control, the word "colour" was replaced with the word "color" which is standard in the software development. As a result, such enumerations and properties as **eilColourDepth**, **SystemColourDepth**, **ColourDepth** were renamed to **eilColorDepth**, **SystemColorDepth**, **ColorDepth**.

A similar replacement was made for the words "image" and "icon" – what we operate is called "image" everywhere. In the context of this change the properties **IconSizeX** and **IconSizeY** were renamed to **ImageWidth** and **ImageHeight** (to mimic the names of the corresponding properties of the standard ImageList control in VB).
19. [Change][Fixed] Some bugs were fixed in the **AddFromPictureBox** method. Its syntax was also simplified – now you use just one parameter (**pic**) to specify the PictureBox you add a picture from without the additional parameter **hdc** in the previous versions.
20. [Fixed] The control might hang if you worked with icons which have more than about 4000 pixels.
21. [Fixed] An error that lockups the Control in the IDE was fixed. Selecting a range of images in the Open Dialog Box in an empty image list caused some type of error – you could get a property window with a "None" Tab and the message "There are no selected objects with property pages."
22. [Fixed] The "<<" and ">>" buttons used to reorder the images in the property page dialog might work improperly - the image to the right of your selection was moved.
23. [Fixed] The original version of the control does not allow to use a one-character string as the image key. The control simply "loses" it after you switched to run-time mode and then again to debug mode. Some customers also encountered a serious GPF error with their VB projects in this case. Other bugs in saving/restoring image keys were also fixed.
24. [Fixed] The control might ignore your changes you had made in the image size and color depth fields.

v2.01 build 0009 | 2003-Jun-05

1. [Change] The original version of **ItemIndex** returns the 1-based index for string keys. Now it is 0-based.
2. [Fixed] In the original version of vbAccelerator ImageList the error '9' ("Subscript out of range") is raised in run-time mode in some systems when you issue the **ItemIndex** method and specify a string key of an image. Notice that this is not the same bug as stated at vbaccelerator.com (earlier versions raises this error if you want to retrieve the index of the first icon - visit http://www.vbaccelerator.com/home/VB/Code/Controls/ImageList/vbAccelerator_Image_List_Control/article.asp).

The problem is that if you specify string keys for icons, the control stores them in a single string which contains the keys separated by the Chr(255) character. When you launch a compiled (exe) application that contains a vbAccelerator ImageList control with defined icon string keys, the control restores the string with icon keys but replaces the Chr(255) character on Chr(&H3F), the question mark character ("?"), and therefore cannot process icon keys properly.

We suppose that this error is caused by the following fact. VB deals with Unicode strings but it stores them in ASCII format in .frm files when you issue the PropertyBag.WriteProperty method (vbAccelerator ImageList uses it to store the string that represents packed icon keys). Depending on the operating system locale settings the Chr(255) character may be restored properly or not in the corresponding Unicode character; in the last case it is replaced on the question mark character.

To fix this error we changed Chr(255) on the Tab character (vbTab or Chr(9)). This character is saved and restored properly regardless of system locale.

3. [Fixed] The **ItemIndex** function in the original version of the control does not work properly with icon string keys which represent numbers ("11", "2222" and so on). The problem is that original **ItemIndex** uses VB standard function IsNumeric to determine the type of its argument and, of course, returns True

for such strings as "11", "2222". We have replaced IsNumeric with our function, fbIsNumeric, which returns proper results:

```
Private Function fbIsNumeric(ByRef vRow As Variant) As Boolean
```

```
    Select Case VarType(vRow)
```

```
        Case 2 To 6, 14, 17 ' Integer, Long, Single, Double, Currency, Decimal, Byte
```

```
            fbIsNumeric = True
```

```
    End Select
```

```
End Function
```

4. [Fixed] If you use Windows XP Visual Styles in VB IDE (by specifying the manifest file for VB6.exe - visit http://www.vbaccelerator.com/home/VB/Code/Libraries/XP_Visual_Styles/Making_VB_Apply_Visual_Styles_at_Design_and_Debug_Time/article.asp for more info), some controls in vbAccelerator ImageList Property Pages are not drawn properly. This is fixed by inserting these controls in VB PictureBox object (this technique is described in the Notes section at http://www.vbaccelerator.com/home/VB/Code/Libraries/XP_Visual_Styles/Using_XP_Visual_Styles_in_VB/article.asp).
5. [Fixed] vbAccelerator ImageList demo crashes in some versions of Windows XP with Visual Styles turned on. This bug has been fixed by invoking the SetErrorMode WinAPI function in the Form_Terminate event. This technique is also described at vbaccelerator.com at http://www.vbaccelerator.com/home/VB/Code/Libraries/XP_Visual_Styles/Preventing_Crashes_at_Shutdown/article.asp.