






DevPower Development Tools – ActiveX Controls

<http://www.devpower.com>

Welcome to the DevPower Development Tools ActiveX Controls Help File v1.1.3

Please check our web site for the latest ActiveX Controls and updates.

The DevPower ActiveX controls can either be installed from within their web page (this requires Internet Explorer 3 or higher) or by downloading a zip file.

Control	Version	License
 Button Bar	V1.0.1	\$59.95 per developer
 Encrypt	V2.0.0	Free
 Flat Button	V3.1.0	Free – source code available for \$40
 News Ticker	V2.0.0	Free
 Transparent Image	V2.8.0	Free
 Tray Icon	V2.1.0	Free
 Web Color	V1.3.2	Free
 Web Version Check	V2.0.0	Free

Contact Information

Copyright © 1997 – 1998 Steve Robbins

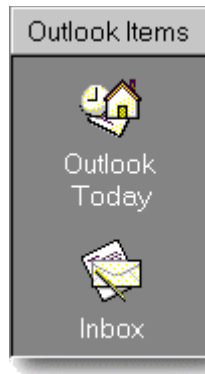
Copyright Information

The DevPower Development Tools are Copyright © 1997 – 1998, Steve Robbins

You are permitted to use the free and trial version DevPower ActiveX Controls as well as distribute the .OCX and associated files with your application royalty-free, provided you do not alter the files or take personal credit for the tools.

Please do not distribute the ZIP files containing the tools on your web site - please link to the files on the DevPower website using the link on each tools' page. This will ensure that the latest version is available to your visitors.

DevPower Button Bar



The Button Bar ActiveX Control lets you easily build a modern user interface like those used in Microsoft Outlook 98 and Project 98. The control provides features such as enabling/disabling of specific tabs and items, control over the colours used and utilisation of Visual Basic collections.

This control is available in two versions; "Button Bar" - the full version of the control that is available for \$59.95 per developer and "Button Bar Lite", which is free, but limited to only displaying 2 tabs and 6 items.

Quoted from [ACTIVEV.COM DISPATCH](http://ACTIVEV.COM/Dispatch), July 7, 1998...

"Itching to borrow some ideas from the user interface of Microsoft Outlook? That Outlook bar sure is handy, as any Outlook user will tell you. Add a similar bar to your own applications with DEVPOWER BUTTNBAR ACTIVEX CONTROL."

Keyboard Navigation

You can let users navigate around the control with the keyboard by setting the TabStop property to True in the design environment. Once the control has focus, the navigation keys are;

- Up / Down arrows to choose items (return to select)
- Page Up / Page Down to scroll items within a tab
- Home / End to move to the top / bottom items in a tab
- Control + Page Up / Page Down to choose tabs
- Control + Home / End to move to the top / bottom tabs.

General Properties

AlwaysShowFirstTab *Boolean*

Whether the control should hide the top tab if only one tab is being used.

BackColor *OleColor*

Sets the background colour of the control.

CheckColor *OleColor*

Sets the color to draw check patterns with.

DkShadowColor *OleColor*

Sets the dark shadow color.

Font *Font*

Sets the font to use for the Tab and Item captions. For details on setting the font in a Web Page, see DevPower FAQs page on our web site.

HighlightColor *OleColor*

Sets the highlight color.

hWnd *Long*

Returns the Window Handle for the control.

ItemsMoveWithClick *Boolean*

Whether images should move down when items are clicked (like a normal button).

ItemsStayDownWithClick *Boolean*

Whether items should toggle down when clicked (like MS Project 98).

ItemTextColor *OleColor*

Sets the Item Caption color.

PlaySounds *Boolean*

Whether the control should play Sounds when items are clicked or the current tab is changed.

ScrollDelay *Integer*

The delay in milliseconds when scrolling the items. The value must be between 10 and 2000, with 150 being the default.

ShadowColor *OleColor*

Sets the shadow color.

ShowToolTips *Boolean*

Whether the control should display tooltips when the mouse pauses over tabs and items.

SmoothScroll *Boolean*

Whether to scroll items and tabs smoothly, like Outlook 98.

SoundItemURL *String*

Sets the URL / file location for the sound to be played when an item is clicked.

SoundTabUpURL *String*

Sets the URL / file location for the sound to be played when a tab is moved up.

SoundTabDownURL *String*

Sets the URL / file location for the sound to be played when a tab is moved down.

Tabs *TabCollection* (Read Only)

Returns the Collection of Tabs through which all tab properties can be set and access to items on tabs can be gained easily.
See Collections below.

TabColor *OleColor*

Sets the color of the Tabs.

TabHeight *Integer*

Returns the current height of Tabs.

TabTextColor *OleColor*

Sets the color of the captions on the Tabs.

UseSystemSounds *Boolean*

Whether the control should play the sounds specified by the user for "Menu Command", "Restore Up" and "Restore Down" or the custom sounds specified by the Sound URL properties.

Tab Properties

NumTabs *Integer* (Read Only)

Gets the number of tabs on the control.

CurrentTab *Integer*

Gets / Sets the currently selected Tab.

CurrentCaption *String*

Gets / Sets the currently selected Tab Caption.

CurrentView *Integer*

Gets / Sets the current Tab View;

0 LargeIcons.

1 SmallIcons.

Item Properties

NumItems *Integer* (Read Only)

Gets the number of items on the current tab.

CurrentItem *Integer*

The current item on the current tab.

CurrentItemEnabled *Boolean*

Whether the current item is enabled.

CurrentItemCaption *String*

The caption for the current item.

CurrentItemMaskColor *OLE_COLOR*

Sets the colour to make transparent on the current item picture.

CurrentItemPicture *Picture*

Sets the bitmap / icon / jpeg/ gif to display (also see URL Picture).

CurrentItemSmallPicture *Picture*

Optionally sets the bitmap / icon / jpeg/ gif to display in Small Icons View.

CurrentItemTag *String*

Gets / Sets any string you need to for each item.

CurrentItemURLPicture *String*

An alternative to the Picture property - sets a URL / file location for a picture.

CurrentItemURLSmallPicture *String*

An alternative to the SmallPicture property - optionally sets a URL / file location for a picture to display in Small Icons View.

General Methods

AboutBox ()

Displays copyright and version information for the control.

RemoveAllTabs ()

Removes all tabs from the control.

Version () *String*

Returns the control version.

Tab Methods

AddTab (*sCaption String*) *Integer*

Adds a new tab with the specified name. Returns the new tab number.

RelabelTab (formally **RenameTab**) (*iTab Integer*) *Integer*

Allows the user to rename the specified Tab. Returns -1 if an invalid Tab was specified or the tab number if the Tab Renaming commences. If the user hits escape, no further action is taken, but if they rename the Tab and hit enter, the TabRenamed event is triggered.

RemoveAllItems (*iTab as Integer*)

Removes all items from the specified tab.

RemoveLastTab () Integer

Removes the last tab from the control. Returns the number of tabs left.

RemoveTab (iTab Integer) Integer

Removes the specified tab from the control. Returns the number of tabs left or -1 if the specified Tab is invalid.

Item Methods

AddItem (iTab Integer, sCaption String, [Optional sURLPicture String]) Integer

Adds an item to the specified Tab, optionally specifying a URL / File Location for the picture to be displayed. Returns -1 if an invalid Tab is specified or the index of the new item.

RelabelItem (formally RenameItem) (iTab Integer, item Integer, sNewCaption String) Integer

Sets the name of the specified item. Returns -1 if the tab / item combination is invalid.

RemoveItem (iTab Integer, item Integer) Integer

Removes the specified item from the specified tab. Returns the number of items left or -1 if the specified tab or item number is invalid.

RemoveLastItem (iTab Integer) Integer

Removes the last item from the specified tab. Returns the number of items left or -1 if the specified tab number is invalid.

Collections

The *Tabs* property of the control returns a TabCollection object (a collection of Tabs - see Visual Basic help for an overview of how to use collections).

Tab Collection Methods

Add (sCaption String [, bbView ViewType, bEnabled as Boolean]) Boolean

Adds a new tab with the specified caption. Returns True if the tab was successfully created.

Count () Long

Returns the number of Tabs in the control.

Item (Index (string or number)) TabObject

Returns the specified tab. This method is the default for the Tab Collection, meaning that it doesn't have to be explicitly called. This simply means that coding `ButtnBar.Tabs.Item("Tab 1")` is the same as `ButtnBar.Tabs("Tab 1")`. Also, the method is overloaded to handle tab references by their caption or their number, for example, you can refer to the first tab (labeled Tab 1) by either `ButtnBar.Tabs("Tab 1")` or `ButtnBar.Tabs(1)`. The TabObject returned is described below.

Remove (Index (string or number)) Boolean

Removes the specified tab. Returns True if the tab was successfully removed.

Tab Object Properties (see above for descriptions)

Caption String

Enabled Boolean

Items ItemCollection

returns the collection of items on a tab - see below

View Boolean

Tab Object Methods

RemoveAllItems ()

Removes all items from the tab.

UserRelabel Integer

Equivalent of RelabelItem Method

Item Collection Methods

Add (sCaption String [, sURLPicture String, bEnabled as Boolean]) Boolean

Adds a new item to the tab with the specified caption. Returns True if the item was successfully created.

Count () Long

Returns the number of Items in the Tab.

Item (Index (string or number)) ItemObject

Returns the specified item. This method is the default for the Item Collection, meaning that it doesn't have to be explicitly called. Again, the method is overloaded to handle item references by their caption or their number. The ItemObject returned is described below.

Remove (Index (string or number)) Boolean

Removes the specified item. Returns True if the item was successfully removed.

Item Object Properties (see above for descriptions)

Caption *String*
Enabled *Boolean*
MaskColor *OLE_COLOR*
Picture *Picture*
SmallPicture *Picture*
Tag *String*
URLPicture *String*
URLSmallPicture *String*

Item Object Methods

UserRelabel *Integer*
Equivalent of RelabelItem Method

Events

Click (*iTab Integer, item Integer*)

Triggered when the control has been clicked. If item = 0 then a tab has been clicked and moved.

ItemRenamed (*iTab Integer, item Integer*)

Triggered if a user successfully renames an Item (after a call to the RelabelItem method).

MouseOver (*iTab Integer, item Integer*)

Triggered when the mouse moves over an Item or Tab. If item = 0 then the mouse is over a Tab and not an Item.

RightClick (*iTab Integer, item Integer*)

Triggered when the control has been right-clicked. If item = 0 then a general area of the tab has been right clicked.

ScrollUp (*iTab Integer*)

Triggered when items are scrolled up on a Tab.

ScrollDown (*iTab Integer*)

Triggered when items are scrolled down on a Tab.

TabRenamed (*iTab Integer*)

Triggered if a user successfully renames a Tab (after a call to the RenameTab method).

DevPower Encrypt



The EnCrypt ActiveX Control lets you easily encrypt and decrypt strings. The control utilises a fast low-security algorithm, making it ideal for simple encryption needs.

Properties

- *None*

Methods

- **AboutBox()**
Displays copyright and version information for the control.
- **EnCrypt**(*String Input {, optional String PassKey }*) *returns String*
Returns the encrypted version of the Input string.
Application-specific security can be obtained by specifying a string (the PassKey argument) that is used in the encryption process instead of using the default key.
- **DeCrypt**(*String Input {, optional String PassKey }*) *returns String*
Returns the decrypted version of the Input string.
- **Version()** *String*
Returns the Control Version.

Events

- *None*

DevPower Flat Button



The FlatBtn ActiveX Control is an incredibly flexible button that can operate in the style of today's popular applications like Internet Explorer, Office97 and Outlook 98 as well as some unique new styles. Usable within a window or in a web page, its flexible interface allows full control over colours, sounds, and behaviour. The source code for this control is available to buy for \$40 from ShareIt Shareware Registration at <http://www.shareit.com/programs/102186.htm> – this allows you to include the power of the Flat Button in your applications without having to distribute the OCX file.

Properties

- **AlignCaption** *Integer*
Sets the position of the caption within the button;
0 0 Align Top.
1 Align Bottom.
2 Align Left.
3 Align Right.
4 Align Middle.
- **AlignPicture** *Integer*
Sets the position of the picture within the button;
0 0 Align Top.
1 Align Bottom.
2 Align Left.
3 Align Right.
4 Align Middle.
- **AutoSize** *Boolean*
Sets whether the control should automatically resize itself to fit around the image or caption it is to display.
- **BackColor** *OleColor*
Sets the background colour of the control.
- **BorderStyle** *Integer*
Sets whether the control has borders;
0 None - no border is displayed when the mouse is moved over the control.
1 FixedSingle - a border is displayed.
- **Caption** *String*
Sets the caption to display on the control. If a '&' symbol is included in the caption, the character following it will be the short cut key for the control.
- **Checked** *Boolean*
Sets whether the control is checked (only if the Style is Check).
- **DroppedDown** *Boolean* (Read Only)
Gets whether the control is currently dropped down.
- **Enabled** *Boolean*
Sets whether the control is enabled.
- **Font** *OleFont*
Sets the font for the caption. For details on setting the font in a Web Page, see DevPower FAQs page on our web site.
- **FontDirection** *Integer*
Sets the direction the font is displayed in (for TrueType fonts only).
0 Normal Text.
1 Downward Text.
2 Reverse Text.
3 Upward Text.
- **hWnd** *Long*
Gets the Window Handle of the control.
- **HighlightColor** *OleColor*
Sets the highlight colour of the control.
- **MaskColor** *OleColor*
Sets the colour to make transparent on the picture.
- **MouseIcon** *OleObject*
Sets the cursor to display when the mouse is over the control. (also see MouseIconURL)
- **MousePointer** *Integer*
Sets one of the standard windows cursors to be displayed when the mouse is over the control (as with Visual Basic Controls);
1 Arrow.
2 Cross (crosshair pointer).

- 3 I beam.
- 4 Icon (small square within a square).
- 5 Size (four-pointed arrow pointing north, south, east, and west).
- 6 Size NE SW (double arrow pointing northeast and southwest).
- 7 Size N S (double arrow pointing north and south).
- 8 Size NW SE (double arrow pointing northwest and southeast).
- 9 Size W E (double arrow pointing west and east).
- 10 Up Arrow.
- 11 Hourglass (wait).
- 12 No Drop.
- 13 Arrow and hourglass.
- 14 Arrow and question mark.
- 15 Size all.
- 99 Custom icon specified by the MouseIcon

- **Picture** *OleObject*
Sets the image to display when the mouse is not on the button (also see URL Picture).
- **PictureDown** *OleObject* (optional)
Sets the image to display when the mouse held down on the button.
- **PictureDisabled** *OleObject* (optional)
Sets the image to display when the control is disabled.
- **PictureOver** *OleObject* (optional)
Sets the image to display when the mouse is over the button.
- **PlaySounds** *Boolean*
Whether to play sounds when the control is clicked or the user moves over / off the button.
- **ShadowColor** *OleColor*
Sets the shadow colour of the control.
- **SoundClick** *Variant (Byte Array)*
Gets / Sets the contents of the sound file downloaded through the SoundClickURL property so that sounds can be shared among buttons at run-time to save download time (also see PropertyDownloaded event).
- **SoundClickURL** *String*
The File Location / URL for the sound to play when the mouse clicks the button (if UseSystemSounds is False).
- **SoundEnter** *Variant (Byte Array)*
Gets / Sets the contents of the sound file downloaded through the SoundEnterURL property so that sounds can be shared among buttons at run-time to save download time (also see PropertyDownloaded event).
- **SoundEnterURL** *String*
The File Location / URL for the sound to play when the mouse moves over the button (if UseSystemSounds is False).
- **SoundLeave** *Variant (Byte Array)*
Gets / Sets the contents of the sound file downloaded through the SoundLeaveURL property so that sounds can be shared among buttons at run-time to save download time (also see PropertyDownloaded event).
- **SoundLeaveURL** *String*
The File Location / URL for the sound to play when the mouse leaves the button (if UseSystemSounds is False).
- **Style** *Integer*
Sets the control style;
0 Normal - a normal button
1 Check - a button that stays down when clicked (like a check box)
2 Separator - a toolbar separator that doesn't respond to user events - this can be used for creating a toolbar with the buttons.
3 DropDown - This allows a side drop down panel on the side of the button and operates like the Office 97 drop down toolbar buttons - e.g. Undo / Redo in Word.
4 DropDownNoSeparator - This drop down style operates like the folder selection button in Microsoft Outlook - the difference with the above style is that there is no separate drop down area, the whole button acts as the drop down selector.
5 Menu - This drop down style operates like the menus used in Office 97 / Windows 98.
6 MenuItem - Operates like the menu items used in Office 97 - these styles can be used for creating menus with the buttons
7 MenuItemCheck - As above but checkable.
8 MenuItemCascade - Menu Item that drops down to have sub menus.
9 MenuItemSeparator - a menu item separator that doesn't respond to user events.
10 NonFlat - Brings the unique features of Flat Button to a normal button.
- **TextColor** *OleColor*
Sets the caption colour.
- **ToolTipText** *String*
Sets the tool tip to display for the control.
- **Transparent** *Boolean*
Sets whether the background of the control is transparent - that is no background color and no rigid border around your image. This feature does not work with icons and the button cannot have a caption, borders and AutoSize must be turned on.
- **URLMouseIcon** *String*
An alternative to the MouseIcon property - sets a File Location / URL for downloading an icon / cursor to display when the

- mouse moves over the button.
- **URLPicture** *String*
An alternative to the Picture property - sets a File Location / URL for downloading an image to display.
- **URLPictureDown** *OleObject* (optional)
See PictureDown
- **URLPictureDisabled** *OleObject* (optional)
See PictureDisabled
- **URLPictureOver** *OleObject* (optional)
See PictureOver
- **UseSystemSounds** *Boolean*
Whether to the user defined sounds for menu command and menu pop-up (if PlaySounds is turned on).
- **Value** *Boolean*
Gets / Sets the value of the button. Setting to True at run time will raise the Click event (same functionality as Visual Basic buttons).

Methods

- **AboutBox** ()
Displays copyright and version information for the control.
- **CancelDropDown** ()
This method should be used to restore the button state after the user has selected an item or cancelled (by clicking away) your drop down menu / window.
- **Version** ()
Returns the control version.

Events

- **Click** ()
Triggered when the control has been clicked.
- **DropDown** ()
Triggered when the user clicks the drop down area of the control.
- **DropDownCancelled** ()
Triggered when the user clicks the button whilst it is in the drop down state - your drop down menu or window should then be hidden.
- **KeyPress** (*Integer KeyAscii*)
Triggered when the user presses a key while the button has focus.
- **MouseDown** (*Integer Button, Integer Shift, Float x, Float y*)
Triggered when a mouse button is pressed down on the control.
Button parameter is a number between 1 and 3.
Shift parameter is a number that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button was pressed or released;
Bit 0 (= 1) - SHIFT key
Bit 1 (= 2) - CTRL key
Bit 2 (= 4) - ALT key
x,y parameters contain the current location of the mouse pointer within the control.
- **MouseMove** (*Integer Button, Integer Shift, Float x, Float y*)
Triggered when the mouse is moved over the control.
Parameters as MouseDown.
- **MouseUp** (*Integer Button, Integer Shift, Float x, Float y*)
Triggered when a mouse button is released over the control.
Parameters as MouseDown.
- **PropertyDownloaded** (*String sProperty*)
Triggered when one of the URL based properties has successfully downloaded a file. This event facilitates the sharing of sounds and images amongst controls meaning reduced download times.
- **RightClick** ()
Triggered when the control has been right-clicked.
- **Resize** (*Long Width, Long Height*)
Triggered when the control resizes itself to fit a picture.

DevPower News Ticker

The NewsTicker ActiveX Control allows you to keep people up to date with latest news in your web site or application. The control allows full control over appearance and behaviour, with properties for background picture, background / highlight / click color, font, etc.

Properties

- **AllowClickWithMove** *Boolean*
Sets whether items can be clicked whilst they are scrolling.
- **BackColor** *OLE_COLOR (Long)*
Sets the background color.
- **Caption** *String*
Sets the list of item captions - a set of strings separated by the Divider character(s) (the divider defaults to a semi-colon).
- **CaptionURL** *String*
Alternative to Caption - specifies a file location / URL for a text file that contains the item captions.
- **ClickColor** *OLE_COLOR (Long)*
Sets the color to display items that have been clicked with.
- **Direction** *(Integer)*
Sets which direction the control scrolls items;
0 – Vertical (default)
1 – Horizontal
- **Divider** *String*
Sets the character(s) that divides the list of item captions (defaults to a semi-colon).
- **Font** *OleFont*
Sets the font for the caption.
- **HighlightColor** *OLE_COLOR (Long)*
Sets the color with which to display items when the mouse is over them.
- **HoverBold** *Boolean*
Sets whether items are displayed in bold when the mouse is over them.
- **HoverUnderline** *Boolean*
Sets whether items are displayed underlined when the mouse is over them.
- **Mouselcon** *OleObject*
Sets the cursor to display when the mouse is over an item.
- **MousePointer** *Integer*
Sets one of the standard windows cursors to be displayed when the mouse is over an item (as with Visual Basic Controls);
1 Arrow.
2 Cross (crosshair pointer).
3 I beam.
4 Icon (small square within a square).
5 Size (four-pointed arrow pointing north, south, east, and west).
6 Size NE SW (double arrow pointing northeast and southwest).
7 Size N S (double arrow pointing north and south).
8 Size NW SE (double arrow pointing northwest and southeast).
9 Size W E (double arrow pointing west and east).
10 Up Arrow.
11 Hourglass (wait).
12 No Drop.
13 Arrow and hourglass.
14 Arrow and question mark.
15 Size all.
99 Custom icon specified by the Mouselcon
- **PauseDelay** *Long*
Sets the time (in milliseconds) to pause scrolling items for reading - defaults to 1000 (1 second).
- **Picture** *OleObject*
Sets the bitmap / jpeg / gif / icon to display as a background for the control (also see PictureURL).
- **PictureURL** *String*
An alternative to the Picture property - sets a URL for downloading a bitmap / icon to display.
- **ScrollDelay** *Long*
Sets the time (in milliseconds) to wait between moving items - defaults to 100 (1/10th second).
- **TextColor** *OLE_COLOR (Long)*
Sets the color of the item text.
- **URLDivider** *String*
Sets the character(s) that divides the item caption from its URL (defaults to =).

Methods

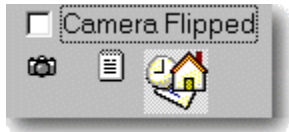
- **AboutBox** ()

- Displays copyright and version information for the control.
Version() *String*
Returns the Control Version.

Events

- **Click** (*Integer itemNumber, String sItem, String sItemURL*)
Triggered when the an item has been clicked.
- **ItemScrollStart** (*Integer itemNumber*)
Triggered before an item starts to scroll.
- **ItemScrollEnd** (*Integer itemNumber*)
Triggered when an item scrolls off of the control.

DevPower Transparent Image



The TransImg ActiveX Control lets you place a bitmaps, icons, jpegs, gifs and metafiles onto a window with a transparent background - a feature that is sadly lacking from many development environments. The transparency acts in two ways; masking out a specific color to the background color and optionally allowing controls behind the image to be seen through the masked area. The control also features scaling of the image, tool tips, properties for downloading the image and cursor from a web server and the option to flip and mirror the image displayed.

Properties

- **AutoSize** *Boolean*
Determines whether the control should automatically resize itself to the size of the bitmap it is to display.
- **BackColor** *OLE_COLOR (Long)*
Sets the background colour to display after the bitmap has been masked.
- **BorderStyle** *Integer*
Sets the controls border style;
0 (None) - Display no border (Default)
1 (FixedSingle) - Display a single pixel border around the control.
- **CheckColor** *OLE_COLOR (Long)*
Sets the check pattern colour (see DrawCheckPattern).
- **DrawCheckPattern** *Boolean*
Sets whether the background of the control has checks on - this is mainly for using this control for rendering buttons.
- **Enabled** *Boolean*
Sets whether the control responds to user actions or not.
- **FlipHoriz** *Boolean*
Sets whether the image is drawn mirrored horizontally.
- **FlipVert** *Boolean*
Sets whether the image is drawn mirrored vertically.
- **hDC** *Long*
Returns the Control Device Context Handle.
- **hWnd** *Long*
Returns the Control Handle.
- **MaskColor** *OLE_COLOR (Long)*
Sets the colour to mask the bitmap with.
- **Mouselcon** *OleObject*
Sets the cursor to display when the mouse is over the control.
- **MousePointer** *Integer*
Sets one of the standard windows cursors to be displayed when the mouse is over the control (as with Visual Basic Controls);
1 Arrow.
2 Cross (crosshair pointer).
3 I beam.
4 Icon (small square within a square).
5 Size (four-pointed arrow pointing north, south, east, and west).
6 Size NE SW (double arrow pointing northeast and southwest).
7 Size N S (double arrow pointing north and south).
8 Size NW SE (double arrow pointing northwest and southeast).
9 Size W E (double arrow pointing west and east).
10 Up Arrow.
11 Hourglass (wait).
12 No Drop.
13 Arrow and hourglass.
14 Arrow and question mark.
15 Size all.
99 Custom icon specified by the Mouselcon
- **Picture** *OleObject*
Sets the bitmap / icon to display (also see URL Picture).
- **Stretch** *Boolean*
Sets whether the displayed image should be stretched to fill the size of the control.
- **ToolTipText** *String*
Sets the tool tip to display for the control.
- **Transparent** *Boolean*

Sets whether the background of the control is transparent - that is no background color and no rigid border around your image. This feature does not work with Icons or at the same time as the Stretch or DrawCheckPattern features.

- **URLMouseIcon** *String*
An alternative to the MouseIcon property - sets a File Location / URL for downloading an icon / cursor to display when the mouse moves over the button.
- **URLPicture** *String*
An alternative to the Picture property - sets a URL for downloading a bitmap / icon to display.

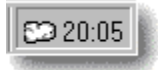
Methods

- **AboutBox** ()
Displays copyright and version information for the control.
- **Cls** ()
Clears the image display.
- **PiSet** (*Long x, Long y, Long Colour*)
Sets a specific pixel to the specified RGB Colour.
- **Point**(*Long x, Long y*)
Returns the RGB Colour of a specific pixel.
- **Refresh** ()
Forces the control to redraw.
- **Version**() *String*
Returns the Control Version.

Events

- **Click** ()
Triggered when the TransImg has been clicked.
- **DblClick** ()
Triggered when the TransImg has been double clicked.
- **MouseDown** (*Integer Button, Integer Shift, Float x, Float y*)
Triggered when a mouse button is pressed down on the TransImg.
Button parameter is a number between 1 and 3.
Shift parameter is a number that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button was pressed or released;
Bit 0 (= 1) - SHIFT key
Bit 1 (= 2) - CTRL key
Bit 2 (= 4) - ALT key
x,y parameters contain the current location of the mouse pointer within the control.
- **MouseMove** (*Integer Button, Integer Shift, Float x, Float y*)
Triggered when the mouse is moved over the control.
Parameters as MouseDown.
- **MouseUp** (*Integer Button, Integer Shift, Float x, Float y*)
Triggered when a mouse button is released over the TransImg.
Parameters as MouseDown.
- **Resize** (*Long Width, Long Height*)
Triggered when the control is resized (usually by the picture changing).
- **RightClick** ()
Triggered when the TransImg has been right clicked.

DevPower Tray Icon



The TrayIcon ActiveX Control lets you easily place an icon in the system tray. This flexible control also allows icons to be downloaded from a web server, meaning it can be used in web pages.

Properties

- **Icon** *OleObject*
Sets the icon to use to display in the system tray. **Use this or the IconURL property.**
- **IconURL** *String*
Sets the URL for the icon to display. **Use this or the Icon property.**
- **Visible** *Boolean*
Sets whether the Tray Icon is displayed in the system tray.
- **ToolTipText** *String*
Sets the Tool Tip to display over the icon when it is displayed in the system tray.

Methods

- **AboutBox**()
Displays copyright and version information for the control.
- **Version**() *String*
Returns the Control Version.

Events

- **Click** ()
Triggered when the TrayIcon has been clicked.
- **DbClick** ()
Triggered when the TrayIcon has been double clicked.
- **MouseDown** (*Integer Button*)
Triggered when a mouse button is pressed over the TrayIcon.
Button parameter is a number between 1 and 3.
- **MouseMove** ()
Triggered when the mouse is moved over the TrayIcon.
- **MouseUp** (*Integer Button*)
Triggered when a mouse button is raised over the TrayIcon.
Button parameter is a number between 1 and 3.
- **RightClick** ()
Triggered when the TrayIcon has been right clicked.

DevPower WebColor



The WebColor ActiveX Control allows you to access the system colors defined by a user in control panel and convert these into HTML color codes for nicer looking forms, tables, etc. Also, the facility to convert HTML color codes to RGB values is provided, which is useful for setting ActiveX control colors.

Properties

- **DefaultColor** *OLE_COLOR (Long)*
Sets the default color to return if the arguments to a method are invalid.

Methods

- **AboutBox** ()
Displays copyright and version information for the control.
- **GetComputerName** () *returns String*
Returns the computer name.
- **GetNamedSysColor** (*String sColorName*) *returns Long*
Returns the RGB value for a named Colour set in control panel. Valid arguments (these aren't case sensitive) are:
 - border, actborder, activeborder, color_activeborder
 - titlebar, activetitlebar, caption, activecaption, color_activecaption
 - app, appworkspace, color_appworkspace, mdi, applicationworkspace
 - bg, background, desktop, color_background, color_desktop, windowbackground
 - 3d, button, btnface, color_btnface, color_3dface, 3dface
 - 3ddkshadow, dkshadow, color_3ddkshadow
 - 3dlight, light, color_3dlight
 - btnhighlight, btnhighlight, color_btnhighlight, color_3dhighlight, color_3dhighlight, color_btnhighlight, 3dhighlight
 - text, btnface, 3dface, color_btnface
 - shadow, btnshadow, color_btnshadow, color_3dshadow, 3dshadow
 - titlebartext, captiontext, color_captiontext
 - titlebargradient, activetitlebargradient, captiongradient, gradientactivecaption, color_gradientactivecaption
 - (Windows 98 / NT5.0 only)
 - inactivetitlebargradient, inactivecaptiongradient, gradientinactivecaption, color_gradientinactivecaption
 - (Windows 98 / NT5.0 only)
 - grayed, disabled, disabledtext, graytext, color_graytext
 - special, highlight, color_highlight
 - specialtext, highlighttext, color_highlighttext
 - hotlight, color_hotlight (Windows 98 / NT5.0 only)
 - inactiveborder, color_inactiveborder
 - inactivetitlebar, inactivecaption, color_inactivecaption
 - inactivetitlebartext, inactivecaptiontext, color_inactivecaptiontext
 - infotext, tooltip, color_infotext
 - infobk, tooltip, tooltipbk, color_infobk
 - mnu, menu, color_menu
 - mnutext, menutext, color_menutext
 - scroll, scrollbar, color_scrollbar
 - wnd, window, color_window
 - wndframe, wndframe, windowframe, color_windowframe
 - wndtext, wndtext, windowtext, color_windowtext
- **GetScreenHeight** () *returns Long*
Returns the screen height.
- **GetScreenWidth** () *returns Long*
Returns the screen width.
- **GetSysColor** (*Long ColorIndex*) *returns Long*
Returns the RGB value for a Colour set in control panel using either the Windows API color index or VB System Color constants.
- **GetUserName** () *returns String*
Returns the user name.
- **HTMLToRGB** (*String sColour*) *returns Long*
Returns the RGB color for an HTML Color code - e.g. HTML of "#FF0000" returns 256.
- **HTMLToRGBValues** (*String sColour, ByRef Integer iRed, ByRef Integer iGreen, ByRef Integer iBlue*) *returns Long*
Returns the RGB color codes for an HTML Color code - e.g. HTML of "#FF0000" returns 256, 0, 0.
- **RGBToHTML** (*Long iColour*) *returns String*
Returns the HTML Color code for an RGB color value - e.g. RGB of 256 returns #FF0000

- **RGBValuesToHTML** (*Integer iRed, Integer iGreen, Integer iBlue*) returns *String*
Returns the HTML Color code for the individual Red, Green and Blue values
- **SetDocBGColor** (*Long IColour*) returns *String*
Adjusts the background color of a web page to the specified colour (providing the hosting web browser supports this).
- **Version** () *String*
Returns the Control Version.

Events

- *None*

DevPower Web Version Check



The WebVrChk ActiveX Control lets you easily add version checking across the internet to your applications. Suggested usage is to periodically (e.g. once a month) call the Check() method that performs a version check for you. If a different version is found the one specified, the control triggers the UpdateAvailable Event. This allows the flexibility to prompt the user to download a newer version if one is available.

Properties

- **AutoLaunchBrowser** *Boolean*
Sets whether the control should automatically launch the users web browser when a new version is found. The browser will display the URL specified in the version file.
- **CurrentVersion** *String*
Sets the version of your application or control to check against.
- **URL** *String*
Sets the URL identifying the WebVrChk control file.
The file contains the following (an example file is available at our web site):
Version=version
- the string to compare with the CurrentVersion property
(Optional) DownloadURL=Any URL
- A URL that will be returned in the UpdateAvailable Event and displayed to the user in their web browser if the AutoLaunchBrowser property is set to True and a new version is found.
(Optional) Abstract=description of changes
- Typically a message to display to users about the new version (however, it can be used however you like) that is returned in the UpdateAvailable Event
(Optional) CustomData=any text data
- Custom information for your use - this field is the only one that can spread across multiple lines in the version file. Again, it is returned in the UpdateAvailable Event

Methods

- **AboutBox()**
Displays copyright and version information for the control.
- **Check()**
Performs the version check - the control attempts to get the file specified by the URL property and parse it for version information as outlined above.

Events

- **UpdateAvailable** (*String NewVersion, String DownloadURL, String Abstract, String CustomData*)
Triggered by the Check() method when a new version has been found in the specified WebVrChk control file.
The *NewVersion* parameter is the version in the control file.
The *DownloadURL* parameter is the optional URL to direct the user to, again specified in the control file. This will be automatically displayed if the AutoLaunchBrowser property is True.
The *Abstract* parameter is the abstract from the control file.
The *CustomData* parameter is the custom data from the control file.
- **ErrorInDownload** (*Integer Error*)
Triggered when an error occurs whilst trying to obtain the control file specified by the URL property.
Error parameter determines the cause of the error:
-1 means 'File not Found' on Web Server.
-2 means 'Could not Connect' to Web Server.

Checking this event means that checking can fail gracefully if the user is not connected to the internet.

Contact Information

Support is available from our web site at <http://support.devpower.com>.

Alternatively, you can email us at help@devpower.com.

You can check for the latest versions of the controls you use at <http://vercheck.devpower.com>.

DevPower Button Bar Registration

Registration fee: \$59.95 per developer.

If you would like to register DevPower Button Bar ActiveX Control, you can do the registration online on the Internet at <http://www.shareit.com/programs/102126.htm>.

Alternatively, you can go to <http://www.shareit.com> and enter the program number there: 102126.

If you do not have access to the Internet, you can register via phone, fax or postal mail. Please print out the registration form, and fax or mail it to:

ShareIt!
element 5 AG
Habsburgerring 3
50674 Koeln
Germany

Phone: +49-221-2407279
Fax: +49-221-2407278
E-Mail: register@shareit.com

US customers may also order by calling 1-800-903-4152 (orders only please!). US check and cash orders can be sent to our US office at

ShareIt! Inc.
PO Box 844
Greensburg, PA 15601-0844
USA

Tel. 724-850-8186
Fax. 724-850-8187

DevPower Button Bar Registration Form

Program No.: 102126

Last name: _____

First name: _____

Company: _____

Street and #: _____

City, State, postal code: _____

Country: _____

Phone: _____

Fax: _____

E-Mail: _____

How would like to receive the registration key/full version?

e-mail - fax - postal mail

How would you like to pay the registration fee of \$59.95:

credit card - wire transfer - EuroCheque - cash

Credit card information (if applicable)

Credit card: Visa - Eurocard/Mastercard - American Express - Diners Club

Card holder: _____

Card No.: _____

Date of Expiration : _____

Date / Signature _____

