

## BALLOON.VBX Custom Control

### Description

The BALLOON Control adds a new technique of providing help to the user that is not available any other way. This control allows the developer to provide the user with a dynamic popup help system for all the controls on each form in the VB application. The popup help text information provided is especially useful to new user and is ideal for custom in-house applications where the user may not have a manual to refer.

When enabled, the control provides a custom event that defines when the user moves the mouse cursor over a form's control. The VB application then passes a help text string to the BALLOON.VBX control that then moves the popup window to the vicinity of the selected control and displays the help text in the popup window.

A separate BALLOON Control must be placed on each form that will have a popup help session available.

A "popup balloon help session" is defined from the time the popup balloon appears on-screen to the point at which the popup balloon window is removed.

### File Name

BALLOON.VBX

### Remarks

When you create and distribute applications that use the BALLOON control, you should install the file BALLOON.VBX in the customer's Microsoft Windows \SYSTEM sub directory. All of the properties, events, and methods for this control are listed below. Properties and events that apply only to this control, or require special considerations when used with it, are underlined. They are documented in this help file. See the Visual Basic *Language Reference* or on-line Help for documentation of the remaining properties, events, and methods.

### Properties

Action

BackColor

BalloonArrow

BalloonHorizontalOffset

BalloonInitBalloon

BalloonLocationMethod

BalloonReferencePoint

BalloonType

BalloonVerticalOffset

BalloonWidth

BalloonWidthMethod

CtlName

FontBold

FontName

FontSize

ForeColor

Index

InverseVideo

Left

MessageText

Tag

Timeout

Top

### **Events**

BalloonClick

BalloonSessionEndClick

### **Methods**

The Action property is used as a pseudo method.

### **Utility Functions**

Included control Utility Funtions are for accessing various control properties.

### **Tips and Techniques**

## Tips and Techniques

### 1. Controls Without hWnd Property:

Some Visual Basic controls do not have hWnd properties. The Visual Basic control types of Image, Shape and Label do not have hWnds. Generally, these type controls do not need help text displayed. Use Text boxes and Picture controls as controls that perform the same functions and have hWnd properties. Any control that does not have a hWnd property will not be highlighted and will not trigger a BalloonClick() event. If any controls without hWnd properties need to have help text displayed, include the control in an enclosing frame that has a hWnd parameter. Use the hWnd of the container control in the Select Case statement of the BalloonClick() event.

### 2. Screen Updates During Balloon Session:

Once the help session has begun, the VISUAL BASIC application MUST NOT take any action or cause any action that would trigger a screen update. For instance; You may NOT send a dialog box to the screen during a help message session. Doing so will cause the help session to immediately terminate, and may leave windows and the VISUAL BASIC application in an unstable state. To properly terminate a help message session, send a Balloon.Action = STOP command to the control. This clears the help session and allows normal application processing to resume. The form's Paint event should not include a DoEvents() statement.

### 3. Terminating A Balloon Session:

Terminating the help message session from within the VISUAL BASIC application can cause the user to become confused. The VISUAL BASIC application should ALWAYS allow a help message session to terminate by the user. A mouse click anywhere on the screen or any keyboard key press ends the current balloon help session.

### 4. Setting Colors At Design Time:

It is important to note that during design time you must explicitly set the BackColor and the ForeColor for each BALLOON.VBX control used in a project. VISUAL BASIC does not have defaults for palette aware controls.

### 5. Setting Colors During Runtime:

If the colors displayed by a BALLOON.VBX control's popup window are altered during runtime, the running application must assume all responsibility for setting the colors properly. The colors DO NOT reset between popup window sessions. If the window containing the BALLOON.VBX control is unloaded, then the colors for only this form will reset to the value set at design time.

### 6. Disabled Form Controls:

Disabled controls return a hWnd to the BalloonClick() event. The application can display help for disabled controls or choose not to. If the VISUAL BASIC application does not want to display help for a disabled control use the Windows API call IsWindowEnabled(hWndAtMouse) to determine whether the control is disabled.

### 7. MDI Applications:

For MDI applications, use a separate control for the MDI form and for each Child form that is to be monitored. The control cannot be placed directly on a MDI form. Place the MDI form's control on a toolbar or status bar.



## **MessageText Property, Balloon Control**

### **Description**

Sets the text to be displayed in the popup window.

The initial text to be displayed is set prior to the issuance of the Balloon1.Action property command. If no default text is supplied, then the control uses the internal message to start a balloon help session.

If no default text is supplied, the control uses the internal message to start a balloon help session.

### **Usage**

`[form.]BALLOON.MessageText[ = setting %]`

### **Data Type**

String, maximum of 255 characters.

### **How to use**

The MessageText property moves the popup help balloon window to the location of the control the mouse is over and puts the text in the popup balloon window.

If the mouse is over a control that help text is NOT desired, then do not set the MessageText property. This "do nothing" default does not move the popup window and does not update the text. The next active control the mouse is placed over will continue the move-update sequence. For example, in the Case Select statement for the window to skip can be omitted.

If the mouse is over a control that help text is NOT desired, do not set the MessageText property.

## **Timeout Property, BALLOON Control**

### **Description**

Setting the Timeout property determines the time the Balloon remains on screen when the mouse remains immobile. When the internal VBX timer reaches the Timeout time with no mouse motion, the help session is terminated and the balloon popup window is removed.

### **Usage**

`[form.]BALLOON.Timeout[ = setting %]`

### **Settings**

The Timeout property settings are in milliseconds.

The minimum setting is 1000 ( 1 second ).

The maximum setting is 32000 milliseconds ( 32 seconds ).

The default setting is 10000 ( 10 seconds ).

### **Remarks**

Setting the Timeout time to 0 disables the Timeout timer function. The balloon will remain onscreen until the user terminates the balloon help session

## Action Property, BALLOON Control

### Description

Setting the Action property causes the BALLOON.VBX control to display on-screen. If the property BALLOON.MessageText was set prior to the issuance of the Action property, then that text is displayed as the first text in the first Balloon. Typically this would be used for introducing the user to the help session. A phrase such as "Move the mouse over the control of interest to get help prompting displayed for it". If no message text is sent prior to the action property, then an internal default message is displayed.

If the property BALLOON.MessageText was set prior to the issuance of the Action property, that text is displayed as the text in the first Balloon.

### Usage

[*form.*]BALLOON.Action[ = *setting %*]

### Settings

The Action property settings are:

Setting	Description
ID_START = 20	Start a popup balloon help session.
ID_STOP = 21	Stop a popup balloon help session.

### Remarks

The code fragment to begin monitoring the form for mouse move events is

```
Form.Balloon1.Action = ID_START
```

The user terminates a popup balloon help session by a mouse click or any keyboard activity. In those rare occasions where it may be necessary to terminate the session from the VB application, then issuing the following command will do so. NOTE: Use caution with this command, as it often results in user confusion, since the program is taking action which effects the program state and the user is not usually notified.

In those rare occasions where it may be necessary to terminate the session from the VB application, issuing the following command will terminate the session.

```
Form.Balloon1.Action = ID_STOP
```

### Data Type

Integer

## **BackColor Property, BALLOON Control**

### **Description**

The background color of the popup window is set by the color palette attached to the control's BackColor property.

The BackColor must be explicitly set to a color; the default is not pre-set when the control is loaded. Each BALLOON.VBX control in a project may have different BackColor properties set.

### **Usage**

`[form.]BALLOON.BackColor[ = setting %]`

### **Settings**

The BackColor property is a color selection palette to choose the popup window's background color. Select the desired color with the mouse, or type in an unsigned long color value.

### **Data Type**

Unsigned long color value. For SDK users, ForeColor uses the same format as the WINDOWS.H COLORREF parameter.

## **BalloonArrow Property, BALLOON Control**

### **Description**

The Balloon Arrow Property of the popup balloon help window is set by selecting true/false from the Properties at design time. The Balloon Arrow Property is a screen graphic that points to the control the help text is generated for. This property allows the designer to customize the screen view of the help balloon.

The properties Balloon Arrow, Balloon Type and Inverse Video may be combined to achieve the desired screen effect.

### **Usage**

Not applicable

### **Settings**

True turns pointer on.

False turns pointer off

### **Data Type**

Boolean

## BalloonInitBalloon Property, WSHELP Control

### Description

The BalloonInitBalloon property of the WSHELP help control is set by selecting true/false from the properties window at design time. The BalloonInitBalloon Property is a screen graphic control that determines whether the first advisory balloon is displayed. Selecting True, displays the user advisory start balloon. Selecting False bypasses the user advisory start balloon.

### Usage

[*form.*]WSHELP1.**BalloonInitBalloon** [= *setting* %]

### Settings

True	displays the user advisory start balloon.
False	bypasses the user advisory start balloon

### Data Type

BOOLEAN

## **BalloonHorizontalOffset Property, BALLOON Control**

### **Description**

The BalloonHorizontalOffset property is the horizontal offset value from the form reference point in pixels. The form reference point for the origin of the offset value (i.e., the 0 value) determined by the setting selected in the BalloonCustomOffset property.

If the BalloonLocationMethod property is set to "Default" this property has no effect.

### **Usage**

[form.]BALLOON1.**BalloonHorizontalOffset** [= *setting* %]

### **Settings**

User supplied value in screen pixels.

### **Data Type**

INTEGER

**See Also:** [BalloonReferencePoint](#), [BalloonLocationMethod](#), [BalloonVerticalOffset](#)

## BalloonReferencePoint Property, BALLOON Control

### Description

The BalloonReferencePoint property determines the balloon reference location for positioning the balloon.

Selection is made by choosing the appropriate value from the enumeration list.

The choices are:

Top/Left uses the balloon top-left for the reference point. Selecting Top/Left also positions the balloon BELOW the point determined by BalloonHorizontalOffset and the BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon top left.

Bottom/Left uses the balloon bottom-left as the reference point. Selecting Bottom/Left also positions the balloon ABOVE the point determined by BalloonHorizontalOffset and BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon bottom left.

Top/Right uses the balloon top-right as the reference point. Selecting Top/Right also positions the balloon BELOW the point determined by BalloonHorizontalOffset and BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon top right.

Bottom/Right uses the balloon bottom-right as the reference point. Selecting Bottom/Right also positions the balloon ABOVE the point determined by BalloonHorizontalOffset and BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon bottom right.

The BalloonReferencePoint location is the apex of the square balloon window at the Top/Left or Bottom/Left corner.

To modify the balloon reference point for different balloon features:

For a balloon with a shadow, the BalloonReferencePoint includes the shadow height of 10 pixels.

For a rounded balloon, the BalloonReferencePoint is the same location as for a square balloon.

For a balloon with the BalloonArrow feature the BalloonReferencePoint is 10 pixels horizontal from the arrow base and 18 pixels vertical from the arrow point.

To calculate the arrow point's horizontal location: If the BalloonReferencePoint property is set to Top/Left or Top/Right add 10 pixels to the horizontal position. If the BalloonReferencePoint property is set to Top/Right or Bottom/Right subtract 10 pixels from the horizontal position.

To calculate the arrow point's vertical location: If the BalloonReferencePoint property is set to Top/Left or Top/Right subtract 18 pixels from the vertical position. If the BalloonReferencePoint property is set to Bottom/Left or Bottom/Right add 18 pixels to the vertical position.

The above reference position modification values are automatically taken into account by the control when setting the final on screen balloon position. The Visual Basic designer should take these values into account when determining the property values for BalloonHorizontalOffset and BalloonVerticalOffset.

If the BalloonLocationMethod property is set to "Default" this property has no effect.

### Usage

[form.]BALLOON1.BalloonReferencePoint [= setting %]

### Settings

0 - Top/Left ...the balloon Top/Left is used for positioning reference.

1 - Bottom/Left...the balloon Bottom/Left is used for positioning reference.

2 - Top/Right ...the balloon Top/Right is used for positioning reference.

3 - Bottom/Right ...the balloon Bottom/Right is used for positioning reference.

**Data Type**

INTEGER

**See Also:** [BalloonLocationMethod](#), [BalloonHorizontalOffset](#), [BalloonVerticalOffset](#)

## BalloonLocationMethod Property, BALLOON Control

### Description

The BalloonLocationMethod property determines how the balloon location is determined. Select the desired property value by choosing from the enumeration list.

Selecting a BalloonLocationMethod other than "Default Location" determines the reference location used to apply the BalloonHorizontalOffset and BalloonVerticalOffset offset value to. The offset values are applied in a linear fashion, both positive and negative. The reference location is defined as the (0,0) point.

If the BalloonLocationMethod property is set to "Default Location" these properties have no effect:

BalloonHorizontalOffset

BalloonVerticalOffset

BalloonReferencePoint

### Usage

[form.]BALLOON1.BalloonLocationMethod [= setting %]

### Settings

0 - Default Location	Use the control's default internal algorithm.
1 - Reference To Control	Offset is referenced to the target controls top, left. Control top left is assumed to be 0,0
2 - Reference To Cursor	Offset is referenced to the cursor's position when the balloon is put on screen.
3 - Reference To Window	Offset is referenced to the parent form's top-left. Form window top-left is assumed to be 0,0.

### Data Type

INTEGER

**See Also:** [BalloonHorizontalOffset](#), [BalloonVerticalOffset](#), [BalloonReferencePoint](#)

## BalloonVerticalOffset Property, BALLOON Control

### Description

The BalloonVerticalOffset property is the vertical offset value from the form reference point in pixels. The form reference point for the origin of the offset value (i.e., the 0 value) determined by the setting selected in the BalloonCustomOffset property.

If the BalloonLocationMethod property is set to "Default" this property has no effect.

### Usage

[form.]BALLOON1.BalloonVerticalOffset [= setting %]

### Settings

User supplied value in screen pixels.

### Data Type

INTEGER

**See Also:** [BalloonReferencePoint](#), [BalloonLocationMethod](#), [BalloonHorizontalOffset](#)

## BalloonWidth Property, BALLOON Control

### Description

The BalloonWidth property sets the width of the balloon in screen pixels. The width value entered does not include the margin around the text and the shadow. Add approximately 10 pixels for the margin, and 10 pixels for the shadow to arrive at the final balloon width.

The BalloonWidth minimum value is 40 pixels. The maximum value is 32000 pixels.

If the BalloonWidthMethod property is set to "Default" this property has no effect.

### Usage

[*form.*]BALLOON1.**BalloonWidth** [= *setting* %]

### Settings

User supplied value in screen pixels.

### Data Type

INTEGER

**See Also:** [BalloonWidthMethod](#)

## BalloonWidthMethod Property, BALLOON Control

### Description

The BalloonWidthMethod property selects the method to use to determine the balloon width. Selection is made by choosing the appropriate value from the enumeration list.

The choices are:

- Use the control's default width calculation algorithm.

- Use the balloon width set in the BalloonWidth property.

If the BalloonWidthMethod property is set to "Default", this property has no effect.

### Usage

[form.]BALLOON1.**BalloonWidthMethod** [= setting %]

### Settings

0 - Default      Use the control's default algorithm.

1 - Custom      Use the pixel value in BalloonWidth to set Balloon Width

### Data Type

INTEGER

**See Also:** [BalloonWidth](#)

## BalloonType Property, BALLOON Control

### Description

The BalloonType Property of the balloon help window is set by choosing the desired type from the enumeration list in the properties window at design time.

The properties BalloonArrow, BalloonType and InverseVideo may be combined to achieve the desired screen effect.

When using the "Type 2: Square balloon 3D with NO drop shadow" style balloon, set the backcolor to light gray. The text color can be set to any value.

### Usage

[form.]WSHELP1.**BalloonType** [= setting %]

### Settings

0 - Square, With Shadow	square balloon with drop shadow.
1 - Square, No Shadow	square balloon with NO drop shadow.
2 - Square, 3D, No Shadow	balloon 3D with NO drop shadow.
3 - Rounded	rounded balloon with NO drop shadow.

### Data Type

INTEGER

## **InverseVideo Property, BALLOON Control**

### **Description**

The Inverse Video Property of the popup balloon help window is set by selecting true/false from the properties window at design time. The Inverse Video Property is a screen graphic control that determines whether the selected control is displayed in Inverse Video when the selection is made.

The properties Balloon Arrow, Balloon Type and Inverse Video may be combined to achieve the desired screen effect.

### **Usage**

Not applicable

### **Settings**

True turns Inverse Video on.

False turns Inverse Video off

### **Data Type**

Boolean

## ForeColor Property, BALLOON Control

### Description

The text color of the popup window is set by the color palette attached to the control's ForeColor property.

The ForeColor must be explicitly set to a color; the default is not pre-set when the control is loaded. Each BALLOON.VBX control in a project may have different ForeColor properties set.

### Usage

*[form.]BALLOON.ForeColor* [= *setting* %]

### Settings

The ForeColor property is a color selection palette to choose the popup window's text color. Select the desired color with the mouse, or type in an unsigned long color (i.e., COLORREF) value.

### Data Type

Unsigned long color value. For SDK users, ForeColor uses the same format as the WINDOWS.H COLORREF parameter.

## FontBold Property, BALLOON Control

### Description

The FontBold property is a standard Visual Basic property. It sets the font bold attribute of the text that appears in the popup window.

### Usage

`[form.]BALLOON.FontBold[ = setting %]`

### Settings

The FontBold property is a standard Visual Basic property. See the manuals for the appropriate values

### Data Type

Boolean (integer).

## FontName Property, BALLOON Control

### Description

The FontName property is a standard Visual Basic property. It sets the font name of the text that appears in the popup window.

### Usage

`[form.]BALLOON.ForeColor[ = font]`

### Settings

The FontName property is a standard Visual Basic property. See the manuals for the appropriate values

### Data Type

String

## FontSize Property, BALLOON Control

### Description

The FontSize property is a standard Visual Basic property. It sets the size of the text that appears in the popup window.

### Usage

`[form.]BALLOON.FontSize [= points]`

### Settings

The FontSize property is a standard Visual Basic property. See the manuals for the appropriate values

### Data Type

Integer.

## BalloonClick Event, BALLOON Control

### Description

This event is generated only after a help session is initiated. A popup help window will appear on-screen. When the user moves the mouse over any control on a form, the control will be displayed in inverse video and the popup window moves to the control windows location, sends the BalloonClick Event to the application and awaits the help text to be applied to the popup help window.

### Syntax

**Sub** *BALLOON\_BalloonClick* ( hwndAtMouse **As Integer** )

### Remarks

Code fragments to illustrate the use of BALLOON Control.

Start up the control;

```
Form.Balloon.MessageText = "Initial message text to display"  
Form.Balloon.Action = ID_START
```

The following SUB processes the returned hwndAtMouse and returns the desired help text to the balloon control;

```
SUB BALLOON1_BalloonClick( hwndAtMouse As Integer )  
    Select Case hwndAtMouse  
        Case Control.hwnd-X1  
            Form.Balloon.MessageText = "message text desired"  
        Case Control.hwnd-X2  
            Form.Balloon.MessageText = "message text desired"  
        .  
        .  
        .  
        Case Control.hwnd-Xn  
            ' No text desired  
            Exit Sub  
        Case Else  
            Exit Sub  
    End Select  
End Sub
```

## **BalloonSessionEndClick Event, BALLOON Control**

Notify event which signals the current Balloon session has been ended. This event can be used to clean up any application state or features that may need resetting or initializing after a Balloon session. For most applications, this event is not used.

## Utility Functions

The Balloon control includes the following utility functions;

**Hwnd2ControlTabIndex:**

Retrieves a control's TabIndex property.

Returns a zero length string if property not found.

**Hwnd2CtlName:**

Retrieves a control's name.

Returns a zero length string if property not found.

**Hwnd2HelpContextID:**

Retrieves a control's HelpContextID.

Returns -1 if property not found.

Only works with VB2 and above.

**Hwnd2TagText:**

Retrieves a control's Tag property.

Returns -1 if property not found.

Each utility function takes a single parameter, the hWnd of the control, for which the property is desired. The declares for using these functions are as follows.

### Hwnd2ControlTabIndex

```
Declare Function Hwnd2ControlTabIndex Lib "BALLOON.VBX" (ByVal hwndControl%) As Integer
```

### Hwnd2CtlName

```
Declare Function Hwnd2CtlName Lib "BALLOON.VBX" (ByVal hwndControl%) As String
```

### Hwnd2HelpContextID

```
Declare Function Hwnd2HelpContextID Lib "BALLOON.VBX" (ByVal hwndControl%) As Long
```

### Hwnd2TagText

```
Declare Function Hwnd2TagText Lib "BALLOON.VBX" (ByVal hwndControl%) As String
```

Place all of the declare statement on a single line.

